2

NAVAL POSTGRADUATE SCHOOL

Monterey, California

THESIS

UPGRADE AND ENHANCEMENT OF THE A.S.
DEPARTMENT FINANCIAL MANAGEMENT INFORMATION
SYSTEM; DEVELOPMENT OF THE FMIS
PROPERTY MANAGEMENT MODULE

by

Thomas Allan Ditri

September, 1991

Thesis Advisor: Tung Bui
Co-Advisor: Shu Liao

Approved for public release; distribution is unlimited

Approved for public release; distribution is unlimited.

Upgrade and Enhancement of the A.S. Department
Financial Management Information System;
Development of the FMIS Property Management Module

by

Thomas Allan Ditri
Lieutenant, United States Navy
B.S., Oregon State University

Submitted in partial fulfillment
of the requirements for the degree of

MASTER OF SCIENCE IN INFORMATION SYSTEMS

from the

NAVAL POSTGRADUATE SCHOOL
September 1991

Author: ______________________
Thomas A. Ditri

Approved by: Tung Bui
Tung Bui, Thesis Advisor

____
Shu Liao, Thesis Co-Advisor

____
David R.Whipple, Chairman
Department of Administrative Sciences

Unclassified
SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a REPORT SECURITY CLASSIFICATION UNCLASSIFIED	1b RESTRICTIVE MARKINGS
2a SECURITY CLASSIFICATION AUTHORITY	3 DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution is unlimited.
2b DECLASSIFICATION/DOWNGRADING SCHEDULE	
4 PERFORMING ORGANIZATION REPORT NUMBER(S)	5 MONITORING ORGANIZATION REPORT NUMBER(S)

6a NAME OF PERFORMING ORGANIZATION Naval Postgraduate School	6b OFFICE SYMBOL *(If applicable)* 54	7a NAME OF MONITORING ORGANIZATION Naval Postgraduate School
6c ADDRESS *(City, State, and ZIP Code)* Monterey, CA 93943-5000		7b ADDRESS *(City, State, and ZIP Code)* Monterey, CA 93943-5000
8a NAME OF FUNDING/SPONSORING ORGANIZATION	8b OFFICE SYMBOL *(If applicable)*	9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c ADDRESS *(City, State, and ZIP Code)*		10 SOURCE OF FUNDING NUMBERS

Program Element No	Project No	Task No	Work Unit Accession Number

11 TITLE *(Include Security Classification)*
UPGRADE AND ENHANCEMENT OF THE A.S. DEPARTMENT FINANCIAL MANAGEMENT INFORMATION SYSTEM ; DEVELOPMENT OF THE FMIS PROPERTY MANAGEMENT MODULE

12 PERSONAL AUTHOR(S) Ditri, Thomas Allan

13a TYPE OF REPORT	13b TIME COVERED	14 DATE OF REPORT *(year, month, day)*	15 PAGE COUNT
Master's Thesis	From To	September, 1991	94

16 SUPPLEMENTARY NOTATION
The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

17 COSATI CODES

FIELD	GROUP	SUBGROUP

18 SUBJECT TERMS *(continue on reverse if necessary and identify by block number)*
FMIS PROPERTY MANAGEMENT MODULE
DBASE IV, VERSION 1.1
DATA-BASED MANAGEMENT SYSTEM

19 ABSTRACT *(continue on reverse if necessary and identify by block number)*

The Administrative Sciences (AS) Department of the Naval Postgraduate School (NPS) maintains a large amount of plant and minor property to support its vast and varied opertations. This property requires accurate record keeping to assure accountability of each item throughout its lifetime, from initial acquisition through disposal. The AS Department implemented a Financial Management Information System (FMIS), through the work of prior NPS students, at the commencement of FY 91. This thesis develops and integrates the Property Management Module into the FMIS to support the management and accountability of the AS Department property. The new expanded version is named FMIS 2.0 An outline covering software maintenance analysis, the Property Management system requirements analysis, and system design methodology is provided. The system was written using dBASE IV, version 1.1 and will transition to operational status from the current FMIS at the beginning of FY 92.

20 DISTRIBUTION/AVAILABILITY OF ABSTRACT
☒ UNCLASSIFIED/UNLIMITED ☒ SAME AS REPORT ☒ DTIC USERS

21 ABSTRACT SECURITY CLASSIFICATION
Unclassified

22a NAME OF RESPONSIBLE INDIVIDUAL	22b TELEPHONE *(Include Area code)*	22c OFFICE SYMBOL
Shu Liao	408-646-2505	ASLC

83 APR edition may be used until exhausted
All other editions are obsolete

SECURITY CLASSIFICATION OF THIS PAGE
Unclassified

ABSTRACT

The Administrative Sciences (AS) Department of the Naval Postgraduate School (NPS) maintains a large amount of plant and minor property to support its vast and varied operations. This property requires accurate record keeping to assure accountability of each item throughout its lifetime, from initial acquisition through disposal. The AS Department implemented a Financial Management Information System (FMIS), through the work of prior NPS students, at the commencement of FY 91. This thesis develops and integrates the Property Management Module into the FMIS to support the management and accountability of the AS Department property. The new expanded version is named FMIS 2.0. An outline covering software maintenance analysis, the Property Management system requirements analysis, and system design methodology is provided. The system was written using dBASE IV, version 1.1 and will transition to operational status from the current FMIS at the beginning of FY 92.

TABLE OF CONTENTS

I. INTRODUCTION

A. BACKGROUND

The need for a computerized data base system for the Administrative Sciences (AS) Department of the Naval Postgraduate School (NPS) has been an ongoing subject of thesis study by several previous NPS students. A basic requirement for the system is to process related resource data in four functional areas: Personnel, Property, Supply, and Travel. The prior theses [Ref. 1,2,3,4] varied in their approaches toward developing a system and culminated in the FY 91 implementation and use of the first version of the Financial Management Information System (FMIS) developed by Neil Ford and Nicholas Zimmon [Ref. 4].

This initial version of the FMIS, developed using dBASE IV, has proven operationally satisfactory. However, it did not include a property management module which is required to track plant and minor property during its lifetime in the AS department. The requirement for this property module was recognized during development of the initial system and was planned to be incorporated as a software maintenance enhancement update through follow-on thesis work. This thesis accomplishes that work and includes a brief overview of software maintenance and the database application development process.

B. FMIS VERSION 2.0

The major change to the FMIS in developing the second version is the integration of a property management system module into the original application. To effectively assimilate the enhancement, the system architecture developed for the initial system was followed as closely as possible. This required close attention to detail during the requirement, evaluation, and design phases of the application development. This ensures that the property management sub-system when completed, could be incorporated into the FMIS. It was crucial to ensure that the common fields needed to link the relations between objects were identical in structure. For these reasons this thesis study does not explore new software tools, instead it concentrates on expanding the existing architecture.

The system was developed using Ashton-Tate's dBASE 4, version 1.1. After extensive personal interviews with the departmental staff, a prototype system was rapidly developed and presented for critique by all expected users. The final property module system incorporated functional and data requirement changes identified during prototype testing. It required programmer coding with the dBASE programming language to obtain the advanced features required of the system as well as procedures required for the successful integration with the original FMIS.

The Property Management sub-system provides new record entry for various fields as illustrated in Appendix E. It also provides retrieval of specific records for editing, deletion of records, and selection of the following property reports:

1. Property Custody Log
2. Property Disposal Report
3. Property Custody History Report
4. Minor Property Inventory Report
5. Plant Property Inventory Report
6. Property Location Report

In addition to its original functions, FMIS version 2.0 now provides an accurate, user-friendly, and efficient means of tracking accountable property custodianship throughout the Administrative Sciences Department.

C. CHAPTER DESCRIPTION

Chapter II reviews basic fundamentals of software maintenance. Operating in a dynamic environment, software must continually be modified in order to perform its required function to meet user satisfaction. These changes can often exceed the effort required to develop the initial system. The type of maintenance in developing FMIS 2.0 will be examined.

Chapter III will review the database application development methodology and outline the methods as used in developing the Property Management Sub-system (PMS). The definition, requirements, evaluation, design, and implementation phases will be covered. The soundness of the new property database relation structure as to which level of normal form it satisfies will be discussed.

A description of all new reports generated by the PMS in FMIS 2.0 will be laid out in chapter IV.

Chapter V, Conclusions, discusses usability of the system and areas for further development. There may be enough perfective maintenance to warrant further changes, possibly as study for another thesis study.

Appendices A through E include sections on requirements documentation, data dictionary, custom programming procedures, application documentation, and a user's guide.

II. SOFTWARE MAINTENANCE

Development of the Property Management sub-system and its subsequent integration into the FMIS program falls under the classification of software maintenance. Maintenance requires a different approach towards development as well as presenting a different set of problems than those that would be encountered when developing an initial system.

A. MAINTENANCE EFFORT REQUIRED

In contrast to the "finished" product of an initial software (s/w) system, maintenance of that system is an ongoing concern. This maintenance effort can easily exceed the entire effort expended on the original project, often exceeding over 60 percent of the total effort exerted on the system throughout its life. Why is there a need for so much maintenance? Rochkind [Ref. 5] provides some insight:

> Computer programs are always changing. There are bugs to fix, enhancements to add, and optimizations to make. There is not only the current version to change, but also last year's version (which is still supported) and next year's version (which almost runs). Besides the problems whose solutions required the changes in the first place, the fact of the changes themselves creates additional problems.

The reason that maintenance is a consistent ongoing effort is that the users are usually never completely satisfied with the product they are using. Additional desired or required features are needed to make the system perform as wanted.

One of the many factors increasing the complexity of s/w maintenance is the turnover of personnel involved in the development of the original system. The time required of the new maintenance programmers to learn the system (the learning curve) is a factor that cannot be underestimated nor overlooked. With all factors considered, changes are often more complex to execute than might appear. The changes incorporated in developing FMIS 2.0 could have been accomplished in considerably less time by the original developers (Ford and Zimmon). The study of code and documentation took approximately a third of the entire maintenance time for the system enhancement. The complete maintenance effort distribution in development of FMIS 2.0 is shown in Figure 2.1. This large percentage of the total effort emphasizes the need for highly accurate and complete system documentation to aid future maintenance efforts.

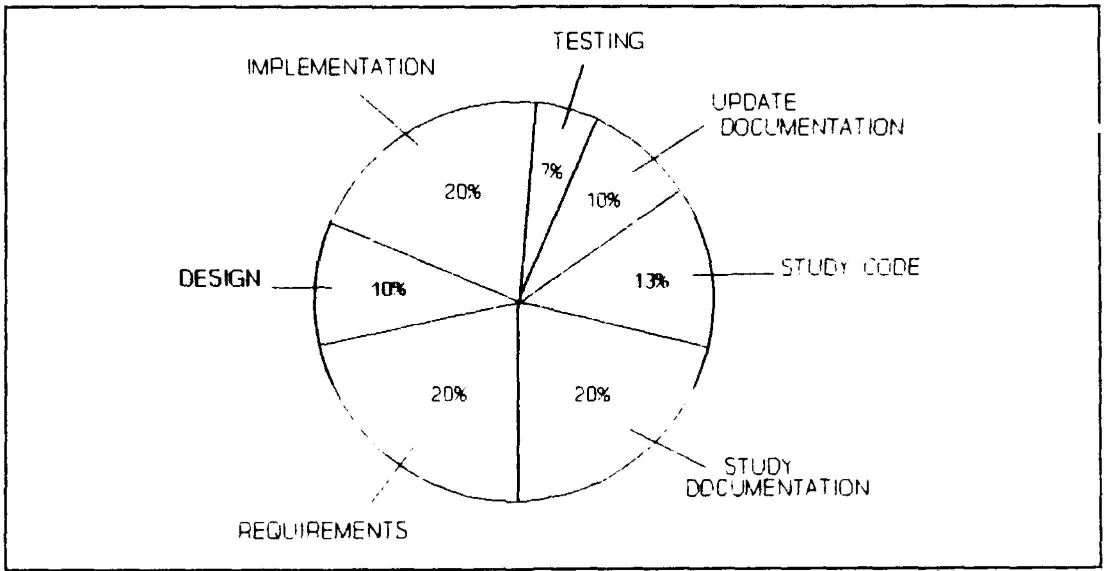


Figure 2.1 FMIS 2.0 Maintenance Effort Distribution

Ensuring a complete software configuration in the original development allows a structured maintenance approach. This is much more efficient than performing unstructured maintenance (maintenance from scratch) which causes a high degree of wasted effort and human frustration. Use of standard dBASE IV configuration (through use of the dBASE control center) in the majority of the initial development provided a sufficient framework for a structured maintenance approach in the upgrade, see Figure 2.2. The only unstructured items to be analyzed were original programming code procedures in the ACCTSPROC.PRG file (see Appendix C) used in the original FMIS.

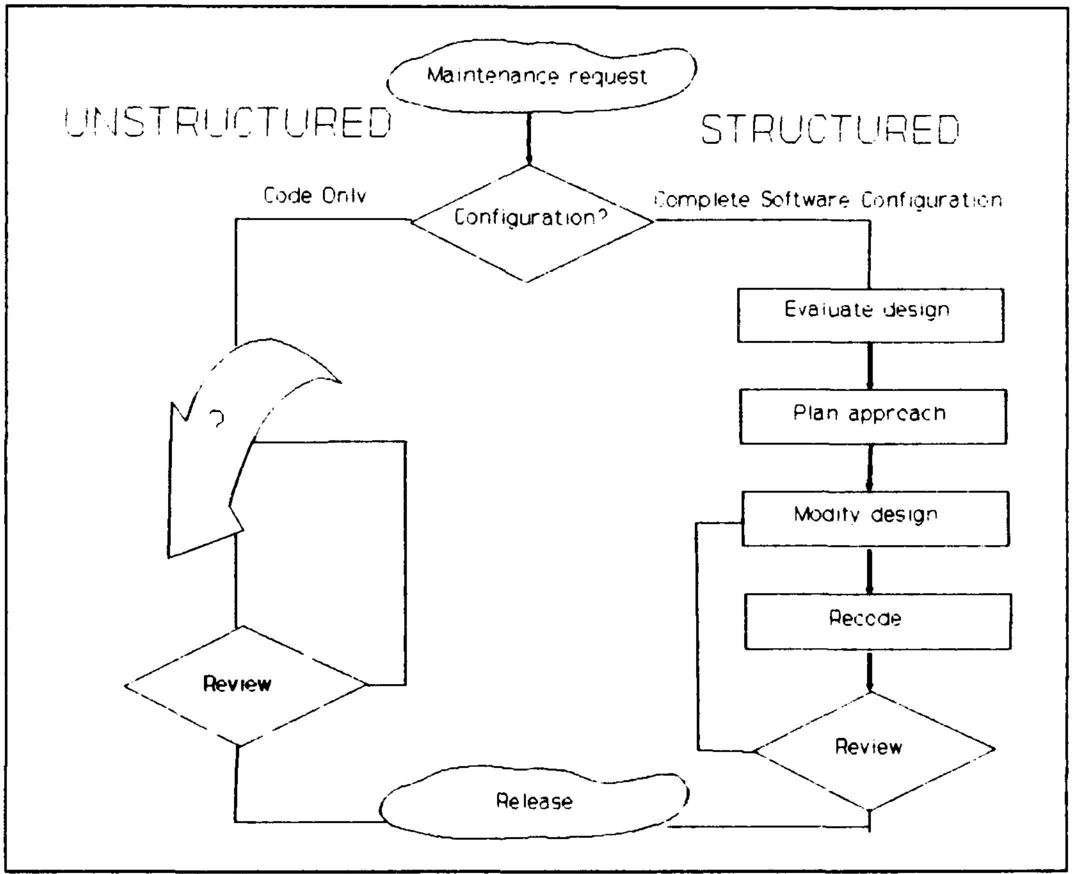


Figure 2.2 Structured Maintenance Approach

B. TYPES OF SOFTWARE MAINTENANCE

There are three major categories of software maintenance as outlined below. Although each type is often interrelated, requiring accomplishment at some or many points during the system lifecycle, each specific maintenance function can easily be categorized.

1. Corrective Maintenance

Corrective Maintenance is the effort involved in correcting errors after initial system delivery. It is practically impossible to discover all errors during system testing. When they eventually occur during system use, they should be recorded by the users and reported to whoever is tasked with maintaining the s/w. The errors must then be diagnosed and corrected. Corrective maintenance usually takes up about 20 percent of the total maintenance effort. No corrective type maintenance was performed on the original system in the development of the new FMIS.

2. Perfective Maintenance

Requiring approximately 10 percent of total maintenance effort, perfective maintenance takes the least amount of programmer attention of the three major categories. This term applies to maintenance performed making improvements of the systems performance and/or quality through modification to existing functions (fine tuning). This is accomplished on systems that have already been proven operationally successful. An example of this might be the reformatting of a report for better readability. About 10 percent of the FMIS 2.0 development was perfective maintenance to the original FMIS.

3. Software Update Maintenance

Update maintenance accounts for the largest chunk of the total maintenance effort, approximately 70 percent. It is divided into two sub-categories as described below.

a. Adaptive Update

This is the work required to modify software to properly interface with an operating environment that may undergo a variety of change. The changing environment may constitute hardware or software reformation. Examples could be either the introduction of a new operating system or hardware upgrades. Adaptive maintenance was not required for the FMIS.

b. Enhancement Update

Enhancements account for about two-thirds of all updates and 45 percent of total maintenance. As successful software is used, requirements for new capabilities beyond the scope of the original system are discovered. Enhancements usually consist of new functional modules to be developed and integrated to the original code.

The Property Management sub-system enhancement accounts for almost all of the work in developing FMIS 2.0.

III. DATABASE APPLICATION DEVELOPMENT

The five phases utilized in the development of the FMIS enhancement will be discussed in this section. Each phase methodology will be discussed followed by how that phase was applied in generating the Property Management sub-system.

A. PHASE I, DEFINITION PHASE

1. Methodology

The definition phase includes preliminary activities with a major goal of simply finding out what needs to be done. The development team must be formed, scope of the project established, and feasibility (cost, technical, and schedule) assessed. This is the simplest phase in the development process.

2. Application

The goal of this project was to develop a Property Management system and integrate it as an enhancement into the FMIS currently in use in the A.S. department office. It was decided that the scope of this work warranted development as an individual thesis project. All feasibility items were met satisfactorily. Work would be performed on a 386, 25 Mhz IBM compatible PC owned by the thesis student. This resulted in negligible cost factors. A time span of eight months with commencement in January 1991 and system completion by August 1991 was considered feasible. Phase I was accomplished during a single interview with Professors Tung Bui and Shu Liao that took approximately one hour.

B. PHASE II, REQUIREMENTS PHASE

1. Methodology

Identifying the objectives of the proposed system in detail is the goal of the requirements phase. Requirements are the blueprint that will be used to design and implement the new system. Before being able to move on to development, the developer must know exactly what the system is supposed to do. It is not only important that the system is built correctly, but vital that the **right** system is built. Proper definition of the requirements can prevent future maintenance nightmares.

There are two major tasks in defining database requirements. The first is to identify the objects. Objects are a collection of properties which depict an item to be implemented in the database. An *object instance* is an example of a specific object. The second step is to determine what functions each application will perform in the database. These requirements are most effectively identified by conducting a series of interviews with the expected users. After initial interviews a prototype may be built and demonstrated to receive further user design input.

2. Application

Interviews commenced the first week of February 1991 with the three expected users of the system: Chan Burnsresearch technician, Jan Evans-administrative officer, and Pearl Murray-Supply clerk. A factor to remember when planning interviews during this phase is that interviewees typically will have full time job duties and schedule interruptions must be expected. Group interviews were beneficial to minimize receipt of conflicting data requirements from the various users.

The initial interviews lasted approximately two and a half weeks. Working with initial data requirements, a prototype Property Management System (PMS) application with sample input screen and reports was developed and presented to the users for review on March 27, 1991. Several changes to the initial requirements were requested by the users and the prototype was reworked with these changes. This cycle was repeated several times over the next two months. Fortunately the time schedule for delivery of the final product had ample flexibility permitting these constant changes.

a. Data Requirements

The PROPERTY object, as determined through the interview/prototype process is shown in the Object Diagram, Table 1, Appendix A. This single new object was the only one required for the system enhancement. All properties of the object listed in the diagram represent an important characteristic of department property items to be tracked. The Personnel property is an *object property*, which means that this entity characteristic is actually another object. The

PERSONNEL object developed in the first FMIS [Ref. 4] will contain properties of the person for whom the department property is assigned custodianship. Additional PROPERTY object data information is supplied in Appendix A, Tables 2 and 3. Table 2 provides the Object definition which lists all of the objects properties and each properties domain. Table 3 is the Domain definition which specifies formats of each domain. This information is used for the database design in Phase IV.

b. Application Functional Requirements

In order to track departmental property items assigned to various custodians, a property clerk must assign a unique tag number to each individual item to be entered into the system. This tag number will identify that particular piece of property entered in the database. Functions required by the PMS were patterned after the existing applications already incorporated into the FMIS. These functions include record entry, display, editing, deletion, and report generation.

The data flow diagram (DFD), Figure 3.1, shows a graphic model of the PMS system to be used as an aid in design. The DFD is comprised of four elements: the data flow, represented by an arrow; the process, represented by a circle; the data store, represented by an open ended rectangle; and the source/sink, which is shown as a closed box.

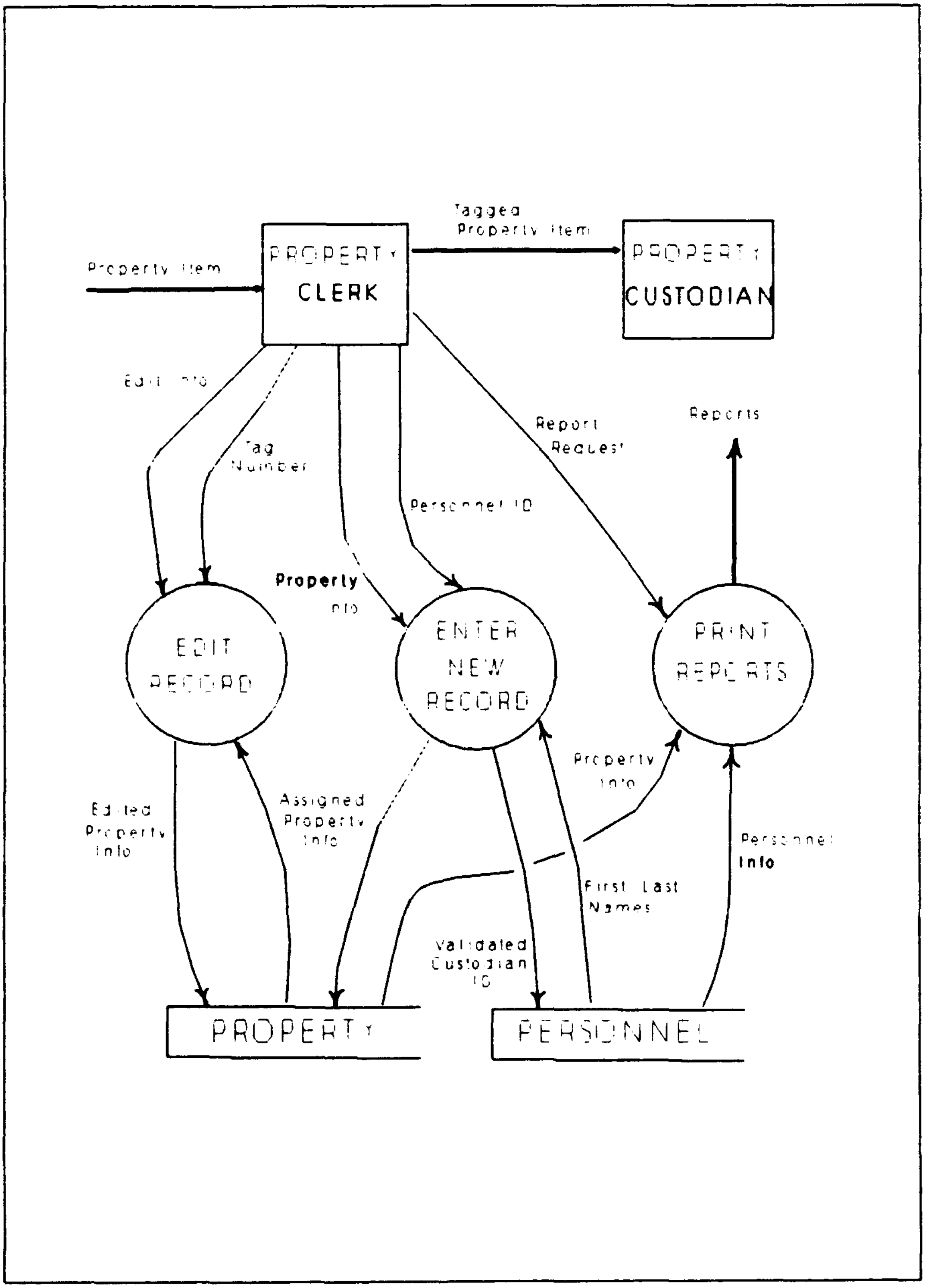


Figure 3.1 Data Flow Diagram

Property items to be tagged and assigned to a staff or faculty member for custodianship are received by the property clerk. Each item ordered under a document number. However, this number cannot be used as a unique identifier since several property items may have been ordered under the same document number. The tag numbers are used as unique property identification. Only one tag number will be assigned tc any one item. Tag numbers are not in strict sequence and may occasionally change depending on the current stock of available tags. Sequences can also vary between minor and plant property.

As shown in Figure 3.1, the ENTER NEW RECORD process requires the input of new property information data along with the custodian's Personnel ID. The process will validate the personnel ID with the PERSONNEL database and retrieve the new custodian's first and last names. The process will not enter a new record with an invalid personnel ID (IDs that do not exist in the PERSONNEL records). The newly assigned property info data record will be stored in the PROPERTY data file.

The EDIT RECORD process requires the entry of a valid tag number. This process searches the PROPERTY data file and retrieves that property record for manipulation or deletion. The edited record will then be stored in the file replacing the original record.

A final process is required to produce the desired reports. Upon receiving a report request, the PRINT REPORTS process will retrieve relevant report data from the PROPERTY and PERSONNEL data files and generate the report. Table 4 and 5, Appendix A, summarize the update and display mechanism requirements for the PROPERTY object.

C. PHASE III, EVALUATION

1. Methodology

Using the information gathered during the requirements phase, this development stage typically consists of an evaluation of several items of concern to the developer and customer.

First is the identification of alternative application system architectures. The question needs to be asked, "Are there other system architectures that would better serve our needs than the one we are planning to use?" Determining the availability of the alternative architecture also needs to be done. Quite often an organization cannot afford the most efficient, state of the art technology.

Another concern that must be evaluated is the feasibility of the project. Can it actually be developed? The detail of the completed requirements may provide insight that preclude further development. Additional time and resources should not be spent on a system that will never be completed. Several large corporations, as well as the government, have abandoned development of poorly evaluated systems after investing tens of millions of dollars. Careful evaluation of the requirements may prevent this from occurring.

A final area of appraisal is the scope of the project. Given the stated time constraints for final delivery, can all functional areas laid out in the requirements be developed? Priorities must be determined and some requirements may need to be postponed and developed at a later date.

2. Application

The Evaluation Phase was simplified by constraints that contribute to the pre-determination of the required architecture as well as a good understanding of the expected requirements during the Definition Phase. As an enhancement to an existing program, it was decided that the new Property Management module would be developed using the same software package (dBASE IV) for ease of integration into the FMIS application. Another factor was the availability of dBASE IV tools on a large number of computers at NPS. The hardware platform was not a consideration either. The system would be run on the AS department office's IBM compatible PC (Northgate 386).

Using the structured maintenance approach it was clear that the planned system upgrade could be accomplished. Feasibility was not a problem. It was evaluated that all requirements as documented in the requirements phase could be completed and delivered on time. This was determined by examining the scope of the original FMIS project (a joint projeci of two NPS students) and estimating the time required for a single student to perform the less extensive enhancement upgrade. One undocumented requirement that needs to be performed at a later date (possible follow-on thesis work) is the development of a security system for the FMIS. A password type system is desired to prevent unauthorized users from gaining access to all FMIS database files.

D. PHASE IV, DESIGN PHASE

1. Logical Database Design

In logical database design, the initial information laid out during the requirements phase will be developed into a set of plans for the database structure. Logical database design is generic, specific design requirements for programming with dBASE IV will be covered by the Physical Database design procedure. The requirements determine what we want and the design determines how to accomplish those goals. Logical design plans developed from the object diagrams and object definitions consist of relation diagrams, relation definitions, and the constraints on the relations.

The PROPERTY object was transformed into the PROPERTY relation as seen in Figure 3.2. The PROPERTY relation uses Property-tag-number (TAGNR) as its primary key. A key is an attribute that functionally determines the non-key attributes. The PROPERTY relation is related to the PERSONNEL relation in a one-to-many binary relaιionship. The "fork" at the PROPERTY

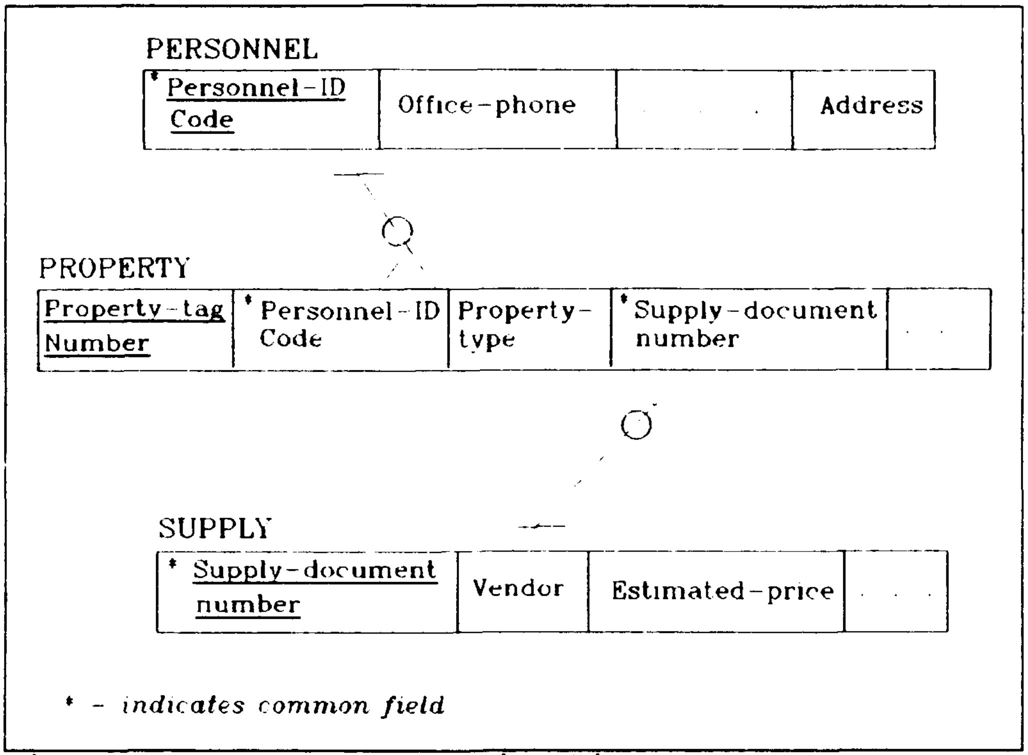


Figure 3.2 Property Relation Diagram

end of the relationship line means that there are potentially many property items for each person in PERSONNEL. The abscense of a fork at the other end indicates that each property item can be assigned to at the most, one person at any one time. The circle on the line means that the relationship from PERSONNEL to PROPERTY is optional. A PERSONNEL member doesn't have to have any property assigned to them. The bar on the line at the other end indicates that a PERSONNEL record must correspond to a PROPERTY record. PROPERTY is linked to SUPPLY in much the same manner.

The relational database model is based on the concept that data is stored in two-dimensional tables referred to as relations. Each row in the table represents a record. Each column represents a field. The entire table (relation) is what is roughly known as a file. A row is called a tuple and a column (field) is called an attribute.[Ref. 6]

The relational structure afforded by dBASE IV allows linking these separate data files through use of a common field. The common field linking PROPERTY and PERSONNEL relations is Personnel-ID-code (IDCODE). The common field linking PROPERTY to SUPPLY is the Supply-document-number (DOCNR). The database hierarchy, updated from the original design [Ref. 4:p. 9], incorporates the new PROPERTY relation as shown in Figure 3.3.

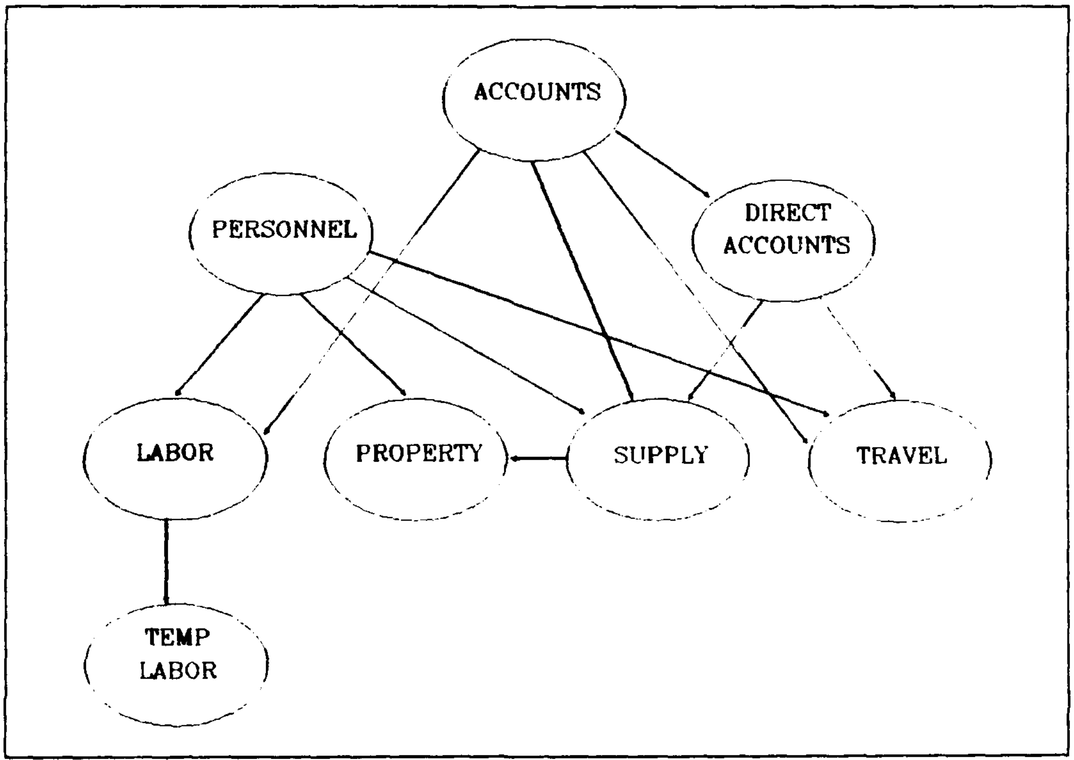


Figure 3.3 Revised Database Hierarchy

2. The Normalization Process

When designing relations from object diagrams, careful attention to the normalization process must be observed to prevent building anomalies into the database structure. Anomalies are weaknesses and flaws in the relations that cause undesirable effects when modifying a database. Types of modification anomalies include deletion and insertion anomalies. Deletion anomalies refer to problems that occur when the deletion of facts from one relation entity inadvertently deletes facts about another entity. Insertion Anomalies describe the restriction of ability to insert information about one entity until additional facts are received about some other entity. Minimization of modification anomalies was of major concern when designing the PROPERTY relation. The "normalization process" is the method to identify and eliminate modification anomalies. Dividing a relation may be required to eliminate any discovered anomalies.

The normalization process consists of testing the relation along a series of normal forms. The term *normal form* refers to the class of relations and techniques for preventing anomalies. There are seven normal forms, the highest level being Domain/Key Normal Form (DK/NF). When a relation is in DK/NF it is guaranteed not to have any anomalies. A relation might fall in any of the normal forms depending on its structure. The normal forms are described below along with the determination of which form requirements the PROPERTY relation satisfies.

a. First Normal Form

The only requirement of this normal form is that the relation has no repeating groups. The PROPERTY relation meets this requirement.

b. Second Normal Form

All non-key attributes must be dependent on all of the key. PROPERTY has a single attribute key (Property-tag-number), therefore it is automatically in second normal form.

c. Third Normal Form

The relation must be in second normal form and have no transitive dependencies. No apparent transitive dependencies could be detected in PROPERTY so it meets this form requirements.

d. Boyce-Codd Normal Form

A relation is in this form if every determinant is a candidate key. However, PROPERTY does not make it past this normal form, a deletion anomaly still exists at this point. If a property item assignment record is removed from the file then the historical information as well as the property details are lost. An alternative approach would be to split the relation into separate PROPERTY-ASSIGNMENT, PROPERTY-DETAIL, and PROPERTY-HISTORY relations. This was not deemed necessary for the PMS system since once a property item is tagged and entered into the file the record should never be deleted. The record is maintained even when the item is disposed. This will prevent the problem of the deletion anomaly from occurring.

e. *Fifth Normal Form*

No clear definition identifies this form, but it is known that even at this level obscure anomalies can occur. This led to the creation of the DK/NF.

f. *Domain/Key Normal Form*

As stated earlier, this is the final form. All possibility of relation modification anomalies have been removed. A relation is in the DK/NF if every constraint on the relation is a logical consequence of the definition of keys and domains [Ref. 6:p. 149].

3. Physical Database Design

This stage of the design phase will transform the logical database design into a physical blueprint to meet specific data element patterns required for programming the application in the dBASE IV dbms. The logical PROPERTY relation attribute names will need to be changed to meet dBASE IV's field name requirement to not exceed 10 characters. The field names must start with an alpha character, but can then be followed by numbers. The underscore (_) is the only non-alphanumeric character allowed in the name. Each field must also be categorized as one of six data types used in dBASE IV:

1. Character - textual information
2. Numeric - any true numeric value

3. Float - useful for numbers with no fixed number of decimal places.

4. Date - date stored in the mm/dd/yy format

5 ogical - contains either a true or false value

6. Memo - large volumes of text (up to 64K)

The total number of fields in the PROPERTY data file numbers 59 (40 of these used for historical data) which easily meets dBASE IV's maximum limitation of 255 fields for any single database record. Table 6 in the Data Dictionary (Appendix B) lists all the PROPERTY data file elements in proper form.

4. Property Management (PMS) Application Design

An application is the collection of menus, forms, reports, and programs that perform the functions of the system required by the users. Before proceeding to the Implementation phase the final task is to design the application. Once the basic designs for the PMS were laid out on paper, a quick prototype was developed to demonstrate the menus, input form screen, and reports to the users. User requested modifications to the prototype were incorporated to form the final application design.

a. Menu Design

Since the development of the PMS system was to be an enhancement to the FMIS already in use, it was decided to follow the current structure in designing the PMS menus. This would allow easier program integration in the implementation phase and ease the user transition to the new system. The menu hierarchy design is illustrated in Figure 3.4. Main menu options from the first FMIS version are shaded. The PROPERTY selection from the main menu produces a pop-up type menu with the selections as shown in the figure. The PRINT REPORTS selection from the pop-up menu produces a pull-down menu with the six various report options. A final pull-down menu (not illustrated) provides the user with choosing either the screen, LPT1, LPT2, or writing to a file as destination options for the chosen report.

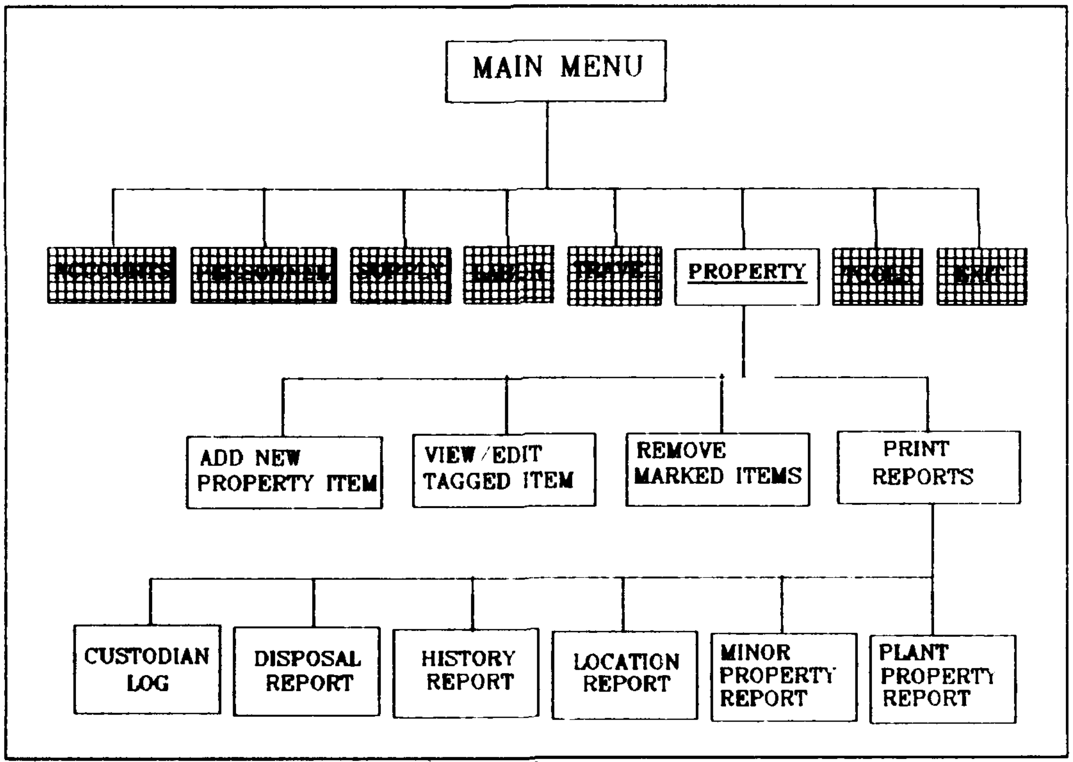


Figure 3.4 PMS Menu Hierarchy

b. Screen design

Variations in the screen layout design can either ease or impede system use. Screen design begins with determination of the information and fields that will be placed the screen and then effectively designing the arrangement so that the data will fit within the screens physical limitations.

The ENTER NEW RECORD and EDIT/VIEW RECORD form screens had to be laid out to meet dBASE IV's physical application design requirements. Custom forms in dBASE can be as wide as the screen, 80 columns. The number of rows, however should be limited to 21 to allow room at the bottom of the screen for pop-up program messages. The screen design for the PMS involves a two page form. The first page consists of all current property and custodian information. The second page contains historical data fields. An entry screen should be designed to minimize the number of keystrokes required by the data entry operator. An efficient design saves entry time and is less frustrating to use. Keeping this in mind, all entries requiring eventual editing were placed at the top of the form. Information required for entry on the second page is automatically transmitted in flashing fields to allow the user to enter historical data without having to refer back to the first page. Another characteristic of good screen design is the varying use of colors. Fields requiring data entry are of a uniform color that is different from information only fields (see Appendix E). This aids in easy identification of entry fields. Another design feature incorporated was the use of default field entry when possible. Several fields identified in the following paragraph automatically fill themselves with predetermined data. This data can be accepted as is, or changed by entering something new.

Various features were desired of certain form screen fields. The following features were designed for the first form page. The TAG NUMBER field was designed to verify that the new number entered had not been previously used in another record. Validation of CUSTODIAN ID with IDs in the PERSONNEL database is required for the form to accept the entry. When a valid ID is entered, the individuals LAST NAME and FIRST NAME fields are automatically filled. Since the majority of AS department personnel utilize offices in Ingersoll Hall, a default entry of "I" is entered for the location room number. The DATE ASSIGNED entry field has a default setting of the current date, entered by the system. ADP CODE has a default entry of "0" for non-ADP property. A default of "M" (minor property) is entered for PROPERTY TYPE. The DISPOSED field gets an automatic value of "N" and also prevents any entry into the DISPOSAL DATE field until it is changed to "Y".

The second page of the form flashes the current CUSTODIAN ID, LOCATION, and DATE ASSIGNED fields transmitted from the first page. These flashing fields cannot be edited which prevents interference of data entered on page one. The final default entries are the first set of historical data fields, CUSTODIAN1, LOCATION1, and ASSIGN DATE1. These are filled with the initial data upon creation of the record.

c. View and Report design

The initial application design prototype consisted of the following four reports: Property Custody Log, Minor Property Inventory, Plant Property Inventory, and the Property History. A final review of the PMS prototype before system implementation resulted in the request for two additional reports, the Property Disposal Record and the Property Location report. A dBASE view file was designed for each report. The view is a representation of a relation using only the fields required for the views intended use. Tables 7 and 8, Appendix B, detail the views and reports.

The basic report designs were drawn up on paper during interviews with the users. Report titles, field locations, groupings, and calculations were decided on at this time. The reports were designed to be printed on 11 by 14 7/8 inch paper which is consistent with reports generated by the original FMIS. As a maintenance enhancement, a major design goal was for the PMS reports to be similar in format to other system application reports. dBASE IV offers three general report layouts: the column layout, the form layout, and the mailmerge layout. The column layout format which includes subtotals and totals was used in all PMS reports.

E. PHASE V, IMPLEMENTATION

1. System Programming

Constructing the system in accordance with the design is the fundamental task of implementation. The control center incorporated by dBASE IV provides most of the tools required for building the various parts of the system. A code generator automatically writes program code to perform each system function constructed in the control center. The database, views, screen form, and reports in the Property Management System must all be created first, then the system can be integrated into the overall FMIS application program. Several special functions and procedures that needed to be coded manually are provided in Appendix C and explained below.

The SRCHTGNR procedure is called by the EDIT/VIEW RECORD property menu option. It searches the PROPERTY database for the record corresponding to the entered tag number and returns either that record or a message stating that the tag number doesn't exist.

PROPOPEN is a procedure that opens the PERSONNE and PROPERTY databases for linked use in the PROPERTY form. This is required to allow simultaneous access of information in both data files. IDCODE is used as a common field between files.

The IsCust function validates the input custodian ID with the PERSONNE file. If valid it returns the corresponding last and first name, if not it produces an error message.

RECRENDX is a procedure to rebuild index files in case they become corrupted by a power failure or some other problem.

The final special procedure, RESTORE, restores the current system drive with a previous backup of all system databases. All of these procedures and functions are contained under one overall procedure file named ACCTPROC.PRG which is opened when the FMIS program is executed.

The dBASE IV application generator provided the tools to make the necessary changes to the original FMIS program and integrate the new PMS module. The sign-on banner was also updated to reflect the new version of the program. Program documentation (Appendix D) was generated as a concluding programming task. Accurate documentation is vital to provide an effective reference for future program maintenance.

2. Testing

Each section of the system was thoroughly checked for correct function using a black box type testing procedure. Black box testing is a testing method where inputs are provided to the system with subsequent checking of the outputs to ensure that the systems overall function is as expected. This testing method does not concern itself with internal functions, rather it checks the correctness of the system as a whole. The PMS system was tested alone and as part of the FMIS 2.0. Approximately 20 various test records were entered into the system for testing. Minor errors were detected and corrected in the entry form and in some of the reports. One of the entry form errors resulted in the necessity of the PROPOPEN procedure discussed in the previous section. A group test was the last stage to see if there were any final functional problems that the programmer may have overlooked. This test went smoothly and any additional changes were agreed to be made as corrective maintenance after implementation.

3. Installation

The final stage of development is Installation. There are two main methods to install a new system. One method is to abandon the old system and start using the new one all at once. The second method, and the one to be used in installing FMIS 2.0, is to run the two systems in parallel. Running the two systems in parallel will allow time for undetected errors in the new system to surface and be corrected before total conversion. This is the preferred method since the original reliable system will still be in place in case of any unexpected problems. The planned time span for parallel operation is 9-30 September, 1991. This is an ideal schedule because the planned conversion will coincide with the start of the new fiscal year (FY 92).

IV. CONCLUSIONS

The initial FMIS has proven itself in actual use through the current fiscal year. FMIS 2.0, a software maintenance enhancement, was the central topic of this thesis. It consisted of the successful development of a Property Management module and integration of this module into the original system. Minor perfective maintenance was also performed on the original system, consisting mainly of report reformatting. FMIS 2.0 is on schedule to replace the original system at the start of fiscal year 92.

The changes incorporated into FMIS 2.0 originated from user requests over the first six months of initial system use. As discussed in section II, software maintenance is an ongoing task throughout system life. Additional desired changes to the program surface constantly, often discovered while performing other maintenance functions. Many of these new requirements and corrections could be incorporated as material for future thesis work. Specific ideas for follow on work are discussed in the following paragraphs.

Within each module in the current system architecture, the same screen form is utilized for both entering and editing records. The development and integration of separate edit forms for each database would alleviate screen congestion during the editing function and also protect permanent data from erroneous changes.

Midway through development of FMIS 2.0 the A.S. department initiated a small microcomputer Local Area Network (LAN). Currently there are only two stations connected, one in I-231 and one in I-231A. The availability of personnel data on this new network brings forth the need for development of database security. As the number of stations expand, access to certain system files must be restricted to authorized users.

Further perfective maintenance can be performed to discover and reduce any remaining anomalies in the system. The goal would be to raise the system to the fourth normal form or higher.

A final suggestion would be to explore the feasibility and benefits of performing an adaptive maintenance change. Specifically, changing the system from dBASE IV to a new software development platform. Alternate database development packages may offer desired features not available in the current system.

APPENDIX A

REQUIREMENTS DOCUMENTATION

A. TABLE 1: OBJECT DIAGRAM

Number
Document-number
Type
Code
Prior-tag
Personnel-ID
PERSONNEL
Model
Acquisition
Date-assigned
Serial-number
Disposal
Disposal-date
Location
Name
Manufacturer
Inventory
Price
Historical-custodian$_{MV}$
Historical-location$_{MV}$
Historical-assignment$_{MV}$
Historical-transfer$_{MV}$

PROPERTY

MV - *Multivalued*

PERSONNEL - *Object Property*

B. TABLE 2: OBJECT DEFINITION

PROPERTY OBJECT

Number; Property-tag-number

Document-number; Supply-document-number

Type; Property-type

Code; Property-ADP-code

Prior-tag; Prior-tag-number

Personnel-ID; Personnel-ID-code

PERSONNEL; PERSONNEL object; SUBSET [First-name,
Last-name]

Model; Model-number

Acquisition; Acquisition-date

Date-assigned; Property-assignment-date

Serial-number; Property-serial-number

Disposal; Disposal-status

Disposal-date; Disposal-date

Location; Property-location

Name; Property-name

Manufacturer; Manufacturer-name

Inventory; Last-inventory-date

Price; Property-cost

Historical-custodian; Past-personnel-IDs; MV

Historical-location; Past-property-locations; MV

Historical-assignment; Past-property-assign-dates; MV

Historical-transfer; Past-transfer-dates; MV

C. TABLE 3: PROPERTY DOMAIN DEFINITIONS

Acquisition-date:
Text 5, mask MM/YY
where MM is month, YY is year
Month and year property item received by supply

Disposal-date:
Date Format (MM/DD/YY)
Date property disposed of

Disposal-status:
Text 1, X
where X is either Y for yes, or N for no
Indicates whether property has been disposed of

Last-inventory-date:
Date Format
Date that last inventory was conducted on property item

Manufacturer-name:
Text 10
Name of property item manufacturer

Model-number:
Text 11
Model number of item (number may include characters)

Past-personnel-IDs:
Text 2
Unique personnel identification code of person having had custody of property item in the past

Past-property-assign-dates:
Date Format
Date property item originally assigned to a past custodian

Past-property-locations:
Text 5, mask B-NNN
where B indicates Building, NNN is room number
Location property items previously assigned

Past-transfer-dates:
Date Format
Date item transferred from past custodian

Personnel-ID-code
Text 2
Unique personnel identification code

TABLE 3 continued

```
PERSONNEL (separate object)
    First-name, Text 10
    Last-name, Text 15
    Name of Property custodian from PERSONNEL file

Prior-tag-number
    Text 9
    Previous identification assigned to item if any

Property-ADP-code
    Numeric, mask X
             where X is either 0 (non-ADP), or 1-5
    Code number to classify item to an ADP category

Property-assignment-date
    Date format
    Date property item assigned to current custodian

Property-cost
    Numeric 12, mask N,NNN,NNN.NN
    Actual cost of property item

Property-location
    Text 5, mask B-NNN
            where B indicates Building, NNN is room number
    Current location of assigned property item

Property-name
    Text 20
    Generic name of property Item, i.e. computer printer

Property-tag-number
    Numeric 6
    Unique AS dept tag number assigned to property item

Property-type
    Text 1, mask M
            where M is either M (Minor) or P (Plant)
    Identifies item as either minor or plant property

Property-serial-number
    Text 15
    Unique number assigned to item by manufacturer

Supply-document-number
    Text 9
    Document number assigned to purchase order of item
```

D. TABLE 4: PROPERTY UPDATE MECHANISMS

I. Add new PROPERTY data
- A. Inputs
  - * Supply information from purchase order
  - * List of available tag numbers
  - * Information of prospective custodian and location
- B. Outputs
  - * New PROPERTY object instance in database
  - * New screen for next record entry
- C. Processing notes
  - * Tag number must be unique
  - * Prospective custodian must exist in PERSONNEL file
  - * Property-cost is actual price paid
- D. Volume
  - * Will vary. Approximately 200 per FY
- E. Frequency
  - * Daily

II. Edit data in PROPERTY
- A. Inputs
  - * Tag number of item to be edited
  - * List of information to be changed
  - * PROPERTY object instance from database
- B. Outputs
  - * Modified object instance to database
- C. Processing notes
  - * Tag number must be valid to enter request
  - * Invalid tag number results in opportunity to try again
  - * Tag number is constant, do not change
- D. Frequency
  - * Semi-weekly

III. Delete PROPERTY data
- A. Inputs
  - * Tag number of record to delete
  - * Information to confirm correct record for deletion
- B. Outputs
  - * Screen message of record deleted
- C. Processing notes
  - * Record for deletion will be retrieved through the Edit process
  - * Records should rarely be deleted since historical property item data is required to be maintained in data file. Delete only erroneous records.
- D. Frequency
  - * Bi-weekly

E. TABLE 5: PROPERTY DISPLAY MECHANISMS

I. Query on PROPERTY
- A. Output description
  - * Form showing all data for a property item
- B. Source data
  - * PROPERTY and PERSONNEL objects
  - * Tag Number keyed by clerk
- C. Processing notes
  - * Used by AS department administrative workers
- D. Volume
  - * 25 per week
- E. Frequency
  - * Daily

II. Minor Property Inventory Report
- A. Output description
  - * Report of Minor property assigned AS tag numbers
- B. Source data
  - * PROPERTY and PERSONNEL objects
- C. Processing notes
  - * Report to be chosen from menu selection
  - * Ordered by tag number
- D. Frequency
  - * Bi-weekly

III. Plant Property Inventory Report
- A. Output description
  - * Report of Plant property assigned AS tag numbers
- B-D. Same as II

IV. Property Custody Log Report
- A. Output description
  - * Report of property items assigned to custodian grouped and sorted by custodian
- B-D. Same as II

V. Property Disposal Report
- A. Output description
  - * Report of disposed property, grouped by ADP code
- B. Source data
  - * PROPERTY object
- C-D. Same as II

VI. Property Location Report
- A. Output description
  - * Report of tagged property grouped by room location
- B-D. Same as II

APPENDIX B

DATA DICTIONARY

A. TABLE 6: PROPERTY.DBF DATA ELEMENTS

ELEMENT	TYPE	WIDTH	DESCRIPTION
DOCNR	CHAR	9	Document number assigned.
TAGNR	NUM	6	AS tag number assigned to property item.
PROPTYPE	CHAR	1	Minor or Plant property identifier (M or P).
ADPCODE	NUM	1	ADP property classification.
PRIORTAGNR	CHAR	9	Previous tag number.
IDCODE	CHAR	2	Personnel identification code.
MODELNR	CHAR	11	Manufacturers model number.
ACQDATE	CHAR	5	MM/YY property received by supply.
ASSIGNDATE	DATE	8	Date property assigned to current custodian.
SERIALNR	CHAR	15	Property item serial number.
DISPOSED	LOGICAL	1	Indicates whether item is disposed of.
DISPDATE	DATE	8	Date of item disposal.
LOCATION	CHAR	5	Location of property item.
REMARKS	CHAR	66	Remarks.
NAME	CHAR	20	Generic name of property item.
DESC	CHAR	20	Descriptive detail of item.
MFG	CHAR	20	Manufacturer name.
INVDATE	DATE	8	Date of latest inventory.
ACTPRICE	NUM	12	Actual price paid for item.

TABLE 6: continued

ELEMENT	TYPE	WIDTH	DESCRIPTION
CUST1	CHAR	2	First person (ID) to have custodianship of property item.
.	.	.	
.	.	.	
.	.	.	
CUST10	CHAR	2	Tenth person (ID) to have custodianship of property item.
LOCATION1	CHAR	5	Location of item during first person assigned as custodian.
.	.	.	
.	.	.	
.	.	.	
LOCATION10	CHAR	5	Location of item during tenth person assigned as custodian.
RECVDATE1	DATE	8	Date item assigned to first custodian.
.	.	.	
.	.	.	
.	.	.	
RECVDATE10	DATE	8	Date item assigned to tenth custodian.
TRANSFER1	DATE	8	Date item transferred from first custodian.
.	.	.	
.	.	.	
.	.	.	
TRANSFER10	DATE	8	Date item transferred from tenth custodian.

Note: CUST2 - CUST9, LOCATION2 - LOCATION9, RECVDATE2-RECVDATE9, and TRANSFER2 - TRANSFER9 field element descriptions omitted from table.

B. TABLE 7: PROPERTY MANAGEMENT SYSTEM VIEWS

VIEW FILE	DATA FILE USED	DATA ELEMENTS
PROPCUST.QBE	PERSONNE.DBF	LASTNAME
	PROPERTY.DBF	IDCODE, TAGNR, LOCATION, NAME, DESC, MFG, MODELNR, SERIALNR, ACQDATE, ACTPRICE, INVDATE, ASSIGNDATE, REMARKS
PROPDISP.QBE	PROPERTY.DBF	ADPCODE, TAGNR, PRIORTAGNR, DISPDATE, NAME, DESC, MFG, MODELNR, SERIALNR, ACTPRICE, REMARKS
PROPHIST.QBE	PROPERTY.DBF	LOCATION1-LOCATION10, TRANSFER1-TRANSFER10, RECVDATE1-RECVDATE10, CUST1-CUST10, TAGNR, NAME, DESC, MFG, MODELNR, SERIALNR
PROPINVM.QBE	PERSONNE.DBF	LASTNAME
	PROPERTY.DBF	TAGNR, LOCATION, NAME, PROPTYPE, DESC, MFG, MODELNR, SERIALNR, IDCODE, ACTPRICE, INVDATE
PROPINVP.QBE	PERSONNE.DBF	LASTNAME
	PROPERTY.DBF	TAGNR, LOCATION, NAME, PROPTYPE, DESC, MFG, MODELNR, SERIALNR, IDCODE, ACTPRICE, INVDATE
PROPLOCN.QBE	PERSONNE.DBF	LASTNAME
	PROPERTY.DBF	IDCODE, TAGNR, LOCATION, NAME, DESC,MFG, MODELNR, SERIALNR,ACQDATE, ACTPRICE, INVDATE,ASSIGNDATE, REMARKS

C. TABLE 8: PMS PROPERTY REPORTS

REPORT FILE	VIEW FILE	DESCRIPTION
PROPCUST.FRG	PROPCUST.QBE	PROPERTY CUSTODY LOG: shows all property items assigned to each custodian. Cost subtotals provided for each custodian.
PROPDISP.FRG	PROPDISP.QBE	PROPERTY DISPOSAL REPORT: shows all disposed property grouped by ADP code classification. Subtotal for each disposed ADP type.
PROPHIST.FRG	PROPHIST.QBE	PROPERTY CUSTODY HISTORY REPORT: all property in tag number sequence showing historical custodian, location, and property assignment date information.
PROPINVM.FRG	PROPINVM.QBE	MINOR PROPERTY INVENTORY REPORT: tag number sequence of all minor property location and custodian (except disposed property).
PROPINVP.FRG	PROPINVP.QBE	PLANT PROPERTY INVENTORY REPORT: tag number sequence of all plant property location and custodian (except disposed property).
PROPLOCN.FRG	PROPLOCN.QBE	PROPERTY LOCATION REPORT: property items grouped in location sequence with subtotals for each location.

APPENDIX C

FMIS 2.0 CUSTOM PROCEDURES IN ACCTPROC.PRG

A. RECRENDX

```
*!**********************************************************
*!  Procedure: RECRENDX
*!             --for re-indexing databases.
*!       Uses: ACCTS.DBF,  DACCTS.DBF,   PERSONNE.DBF
*!             LABOR1.DBF, PROPERTY.DBF, SUPPLY.DBF
*!             TRAVEL.DBF, TEMPLAB.DBF
*!
*!  MDX files: ACCTS.MDX,  DACCTS.MDX,   PERSONNE.MDX
*!             LABOR1.MDX, PROPERTY.MDX, SUPPLY.MDX
*!             TRAVEL.MDX, TEMPLAB.MDX
*!
*!  -This procedure updated to include Property objects for
*!   FMIS version 2.0, by T. Ditri.
*!**********************************************************


PROCEDURE RECRENDX

     SET TALK ON && Show progress
     USE accts
     REINDEX
     USE daccts
     REINDEX
     USE personne
     REINDEX
     USE labor1
     REINDEX
     USE supply
     REINDEX
     USE travel
     REINDEX
     USE templab
     REINDEX
     USE property
     REINDEX
     SET TALK OFF && Suppress progress messages

RETURN
```

B. RESTORE

```
*!***********************************************************
*!  Procedure: RESTORE
*!            --Procedure for restoring backed-up dbase files
*!        Uses: ACCTS.DBF,  DACCTS.DBF,   PERSONNE.DBF
*!              LABOR1.DBF, PROPERTY.DBF, SUPPLY.DBF
*!              TRAVEL.DBF, TEMPLAB.DBF
*!
*!  MDX files: ACCTS.MDX,  DACCTS.MDX,   PERSONNE.MDX
*!              LABOR1.MDX, PROPERTY.MDX, SUPPLY.MDX
*!              TRAVEL.MDX, TEMPLAB.MDX
*!
*!  -This procedure updated to include Property objects for
*!   FMIS version 2.0, by T. Ditri.
*!***********************************************************

PROCEDURE restore

     SET TALK ON && Show progress
     SET SAFETY OFF
     USE accts
     ZAP
     APPEND FROM a:\accts
     REINDEX
     USE daccts
     ZAP
     APPEND FROM a:\daccts
     REINDEX
     USE personne
     ZAP
     APPEND FROM a:\personne
     REINDEX
     USE labor1
     ZAP
     APPEND FROM a:\labor1
     REINDEX
     USE supply
     ZAP
     APPEND FROM a:\supply
     REINDEX
     USE travel
     ZAP
     APPEND FROM a:\travel
     REINDEX
     USE templab
     ZAP
     APPEND FROM a:\templab
     REINDEX
     USE property
     ZAP
     APPEND FROM a:\property
     REINDEX
     SET TALK OFF && Suppress progress messages

RETURN
```

C. SRCHTGNR

```
*!**********************************************************
*!      Procedure: SRCHTGNR
*!               --validates tag number and retrieves record
*!      Calls: PROPERTY.FMT
*!      Uses: PROPERTY.DBF
*!
*!      MDX files: PROPERTY.MDX
*!      Formats: PROPERTY.FMT
*!
*!      This procedure is part of FMIS 2.0, by T. Ditri
*!**********************************************************

PROCEDURE SRCHTGNR
*---Property database opened through embedded code in the
*   application program
     SET TALK OFF
     SET ESCAPE OFF
     SET STATUS OFF
     SET SCOREBOARD OFF
     SET NEAR ON
     searching = .T.

     DO WHILE searching
          CLEAR
          memtag = 0
          @10,2 SAY "Enter Item Tag Number to retrieve: " GET;
          memtag

          READ

          *---If nothing entered, retrieve first record for
          *   browsing.
          IF memtag = 0
               SET FORMAT TO property
               EDIT NOAPPEND
               CLOSE FORMAT
               Searching = .F.
               loop
          ENDIF
          *---Try to find that Tag number.
          SEEK (memtag)

          IF FOUND()
               SET FORMAT TO property
               EDIT NOAPPEND
               CLOSE FORMAT
               searching = .F.
               LOOP

          ELSE
               @12,2 SAY CHR(7)+;
               "Sorry, Tag Number is not in database. Try;
               again." COLOR R/W
               WAIT
          ENDIF
     ENDDO
RETURN
```

D. PROPOPEN

```
*!**********************************************************
*!       Procedure:  PROPOPEN
*!                --opens & links PERSONNE and PROPERTY files
*!       Calls: PROPERTY.FMT
*!       Uses: PROPERTY.DBF
*!             PERSONNE.DBF
*!       MDX files: PROPERTY.MDX, PERSONNE.MDX
*!       Formats: PROPERTY.FMT
*!
*!       This procedure is part of FMIS 2.0, by T. Ditri
*!**********************************************************

PROCEDURE PropOpen
     SELECT A
     USE Property ORDER Tagnr
     SELECT B
     USE Personne ORDER IDcode

     *** Set up relationship based on key fields.
     SELECT Property
     SET RELATION TO IDcode INTO Personne
     GO TOP
RETURN
```

E. ISCUST

```
*!**********************************************************
*!      Function:  IsCust
*!               --validates custodian ID and retrieves last
*!                 and first name for the property screen.
*!      Calls: PROPERTY.FMT
*!      Uses: PERSONNE.DBF
*!
*!      MDX files: PERSONNE.MDX
*!      Formats: PROPERTY.FMT
*!
*!      This procedure is part of FMIS 2.0, by T. Ditri
*!**********************************************************

FUNCTION IsCust
PARAMETERS CustId
        DO CASE

           **If user is exiting, do nothing.
           CASE CustId = " "
                Ok=.T.

           **Personnel ID code was entered
           CASE SEEK(CustId,"Personne")
                @ 3,35 SAY Personne->Lastname
                @ 3,52 SAY Personne->Firstname
                Ok=.T.
           OTHERWISE
                @ 3,35 SAY "No such ID code"
                @ 3,52 SAY SPACE(10)
                Ok=.F.
        ENDCASE
RETURN (Ok)
```

APPENDIX D

FMIS 2.0 APPLICATION DOCUMENTATION

```
Application Documentation for System:  RMS.PRG
Application Enhancement Author:  LT T.A. Ditri, USN
Original Application Authors: LCDR N.S. Ford and LT N.W.
                              Zimmon
dBASE IV Version.....: 1.1

Display Application Sign-On Banner: Yes

Screen Image:
     10        20        30        40        50        60
     >.....+....¦....+....¦....+....¦....+....¦....+....¦
00:
01:
02:
03:╔════════════════════════════════════════════════════╗
04:║                                                    ║
05:║                                                    ║
06:║                                                    ║
07:║   Welcome to the Administrative Science Dept's     ║
08:║                                                    ║
09:║     Financial Management Information System        ║
10:║                     *FMIS*                         ║
11:║                                                    ║
12:║                    Ver. 2.0                        ║
13:║                                                    ║
14:║                                                    ║
15:║                                                    ║
16:╚════════════════════════════════════════════════════╝
17:
Main Menu to Open after Sign-On: RMSMAIN.BAR

Sets for Application:
---------------------
  Bell         ON
  Carry        OFF
  Centry       OFF
  Confirm      OFF
  Delimiters   OFF
  Display Size 25 lines
  Drive
  Escape       ON
  Path
  Safety       ON
```

```
Starting Colors for Application:
--------------------------------
 Color Settings:
   Text             : W+/B
   Heading          : W+/B
   Highlight        : GR+/BG
   Box              : GR+/R
   Messages         : W+/B
   Information      : B/W
   Fields           : N/BG

Database/View: ACCTS
Index Order: JON

Layout Report for Horizontal Bar Menu: RMSMAIN
----------------------------------------------

Screen Image:

0         10        20        30        40        50
>....+....¦....+....¦....+....¦....+....¦....+....¦
00:
╔═══════════════════════════════════════════════════
║ACCOUNTS      PERSONNEL    SUPPLY   LABOR   TRAVEL
╚═══════════════════════════════════════════════════
04:
        60        70
....+....¦....+....¦....+.
══════════════════════════╗
PROPERTY    TOOLS    EXIT ║
══════════════════════════╝

Setup for RMSMAIN follows:
--------------------------
Colors for Menu/Picklist:
-------------------------
 Color Settings:
   Text             : W+/N
   Heading          : W+/N
   Highlight        : GR+/B
   Box              : W+/R
   Messages         : W+/N
   Information      : B/W
   Fields           : N/BG

Before Menu dBASE Code RMSMAIN:
-------------------------------
 SET PROCEDURE TO ACCTPROC
```

```
Bar actions for Menu RMSMAIN follow:
------------------------------------
Bar: 1
 Prompt: ACCOUNTS

 Action: Open a Popup Menu Named: ACCTMENU
------------------------------------------------------------

Bar: 2
 Prompt: PERSONNEL
 Action: Open a Popup Menu Named: PERSMENU
------------------------------------------------------------

Bar: 3
 Prompt: SUPPLY
 Action: Open a Popup Menu Named: SUPPMENU
------------------------------------------------------------

Bar: 4
 Prompt: LABOR
 Action: Open a Popup Menu Named: LABMENU
------------------------------------------------------------

Bar: 5
 Prompt: TRAVEL
 Action: Open a Popup Menu Named: TRAVMENU
------------------------------------------------------------

Bar: 6
 Prompt: PROPERTY
 Action: Open a Popup Menu Named: PROPMENU
------------------------------------------------------------

Bar: 7
 Prompt: TOOLS
 Action: Open a Popup Menu Named: TOOLMENU
------------------------------------------------------------

Bar: 8
 Prompt: EXIT
 Action: Open a Popup Menu Named: EXITMENU
------------------------------------------------------------
```

```
Layout Report for Popup Menu: ACCTMENU
--------------------------------------
Screen Image:
   0         10        20        30        40        50
>....+....|....+....|....+....|....+....|....+....|....+...
03:
04:
05:   ╔═══════════════════════════╗
06:   ║ADD NEW ACCOUNTS           ║
07:   ║VIEW/EDIT ACCOUNTS         ║
08:   ║REMOVE MARKED ACCOUNTS     ║
09:   ║PRINT EXPENSE SUMMARY      ║
10:   ║PRINT OTHER LABOR REPORT   ║
11:   ║---------------------------║
12:   ║DIRECT FUND ALLOCATION     ║
13:   ╚═══════════════════════════╝
14:

Setup for ACCTMENU follows:
---------------------------
 Use database/view and index file(s) in effect at run time.

Colors for Menu/Picklist:
-------------------------
 Color Settings:
   Text             : W+/N
   Heading          : W+/N
   Highlight        : GR+/B
   Box              : W+/R
   Messages         : W+/N
   Information      : B/W
   Fields           : N/BG

Before Menu dBASE Code ACCTMENU:
--------------------------------
 SET STATUS OFF
 SET SCOREBOARD OFF

Bar actions for Menu ACCTMENU follow:
-------------------------------------
 Prompt: ADD NEW ACCOUNTS
 Action: APPEND
 Format File: accts.fmt

 Before dBASE Code for this item:
 --------------------------------
   *---Open Accounts database
   USE PERSONNE ORDER PI
   USE ACCTS
 After dBASE Code for this item:
 --------------------------------
   *---Close Accounts database
   CLOSE DATABASES
------------------------------------------------------------

Bar: 2
 Prompt: VIEW/EDIT ACCOUNTS
 Action: Run dBASE Program: DO SRCHJON

 Before dBASE Code for this item:
 --------------------------------
   *---Open Accounts database
   USE PERSONNE ORDER PI
   GO TOP

 After dBASE Code for this item:
 --------------------------------
   *---Close Accounts database
   CLOSE DATABASES
------------------------------------------------------------

Bar: 3
 Prompt: REMOVE MARKED ACCOUNTS
 Action: Pack Current File
 Window WINDOW1 FROM 10,10 TO 20,60 Double

 Before dBASE Code for this item:
 --------------------------------
   *---Pack Accounts database
   USE accts

 After dBASE Code for this item:
 --------------------------------
   *---Save changes and close database
   CLOSE DATABASES
------------------------------------------------------------

Bar: 4
 Prompt: PRINT EXPENSE SUMMARY
 Action: Run dBASE Program: DO EXPSUM
 After dBASE Code for this item:
 --------------------------------
   set print off
   close databases
------------------------------------------------------------

Bar: 5
 Prompt: PRINT OTHER LABOR REPORT
 Action: Run Report Form OTHERPAY.frm
 Command Options:
  PLAIN
  NOEJECT
 Print Mode: Send to Default Printer
 New Database/View: OTHERPAY.QBE

 After dBASE Code for this item:
 -------------------------------
   set console on
------------------------------------------------------------

Bar: 6
 Prompt: --------------------------
 Action: Text only defined for this option - NO ACTION
------------------------------------------------------------

Bar: 7
 Prompt: DIRECT FUND ALLOCATION
 Action: Open a Popup Menu Named: DACCMENU
------------------------------------------------------------

Layout Report for Popup Menu: PERSMENU
-------------------------------------
Screen Image:
0          10         20         30         40         50
>....+....|....+....|....+....|....+....|....+....|....+...
04:
05:                  ╔══════════════════════════════╗
06:                  ║ADD NEW PERSONNEL             ║
07:                  ║VIEW/EDIT PERSONNEL           ║
08:                  ║REMOVE MARKED PERSONNEL       ║
09:                  ║PRINT PERSONNEL REPORT        ║
10:                  ║PRINT APPT STATUS REPORT      ║
11:                  ║PRINT 30 DAY APPT STATUS REPORT║
12:                  ╚══════════════════════════════╝

Setup for PERSMENU follows:
---------------------------
 Use database/view and index file(s) in effect at run time.

Colors for Menu/Picklist:
-------------------------
 Color Settings:
   Text            : W+/N
   Heading         : W+/N
   Highlight       : GR+/B
   Box             : W+/R
   Messages        : W+/N
   Information     : B/W
   Fields          : N/BG
```

```
Bar actions for Menu PERSMENU follow:
-------------------------------------
Bar: 1
 Prompt: ADD NEW PERSONNEL
 Action: APPEND
 Format File: personne.fmt

 Before dBASE Code for this item:
 --------------------------------
   *---Open Personnel database
   USE PERSONNE

 After dBASE Code for this item:
 --------------------------------
   *---Close Personnel database
   CLOSE DATABASES
------------------------------------------------------------

Bar: 2
 Prompt: VIEW/EDIT PERSONNEL
 Action: Run dBASE Program: DO SRCHPER

 After dBASE Code for this item:
 --------------------------------
   *---Close Personnel database
   CLOSE DATABASES
------------------------------------------------------------

Bar: 3
 Prompt: REMOVE MARKED PERSONNEL
 Action: Pack Current File
 Window WINDOW3 FROM 10,10 TO 20,60 Double
 Before dBASE Code for this item:
 --------------------------------
   *---Open Personnel database
   USE personne

 After dBASE Code for this item:
 --------------------------------
   *---Save changes and close database
   CLOSE DATABASES
------------------------------------------------------------

Bar: 4
 Prompt: PRINT PERSONNEL REPORT
 Action: Run Report Form PERSON.frm
 Command Options:
  PLAIN
  NOEJECT
 Print Mode: Send to Default Printer
 New Database/View: PERSON.QBE
 After dBASE Code for this item:
 --------------------------------
   SET CONSOLE ON
------------------------------------------------------------

Bar: 5
 Prompt: PRINT APPT STATUS REPORT
 Action: Run Report Form APPSTATU.frm
 Command Options:
  PLAIN
  NOEJECT
 Print Mode: Send to Default Printer
 New Database/View: APPSTATU.QBE

 After dBASE Code for this item:
 --------------------------------
   SET CONSOLE ON
------------------------------------------------------------

Bar: 6
 Prompt: PRINT 30 DAY APPT STATUS REPORT
 Action: Run Report Form APPST30.frm
 Command Options:
  PLAIN
  NOEJECT
 Print Mode: Send to Default Printer
 New Database/View: APP30ST.QBE
 After dBASE Code for this item:
 --------------------------------
   SET CONSOLE ON
------------------------------------------------------------

Layout Report for Popup Menu: SUPPMENU
--------------------------------------

Screen Image:
    10        20        30        40        50
+....¦....+....¦....+....¦....+....¦....+....¦....+....¦....+....
04:
05:                    ╔═══════════════════════════════╗
06:                    ║ADD NEW TRANSACTIONS           ║
07:                    ║VIEW/EDIT TRANSACTIONS         ║
08:                    ║REMOVE MARKED TRANSACTIONS     ║
09:                    ║PRINT AGING REPORT             ║
10:                    ║PRINT OUTSTANDING REQN REPORT  ║
11:                    ║PRINT SUPPLY OBLIGATION REPORT ║
12:                    ║PRINT REQN STATUS REPORT       ║
13:                    ╚═══════════════════════════════╝
14:
```

```
Setup for SUPPMENU follows:
---------------------------
 Use database/view and index file(s) in effect at run time.

Colors for Menu/Picklist:
-------------------------
 Color Settings:
   Text            : W+/N
   Heading         : W+/N
   Highlight       : GR+/B
   Box             : W+/R
   Messages        : W+/N
   Information     : B/W
   Fields          : N/BG

Bar actions for Menu SUPPMENU follow:
-------------------------------------
Bar: 1
 Prompt: ADD NEW TRANSACTIONS
 Action: APPEND
 Format File: supply.fmt
 Before dBASE Code for this item:
 --------------------------------
   *---Open Supply database
   SELECT A
   USE PERSONNE ORDER PI
   SELECT B
   USE ACCTS ORDER JON
   SELECT C
   USE SUPPLY
   SET RELATION TO PI INTO PERSONNE, JON INTO ACCTS

 After dBASE Code for this item:
 --------------------------------
   *---Close Supply database
   CLOSE DATABASES
------------------------------------------------------------

Bar: 2
 Prompt: VIEW/EDIT TRANSACTIONS
 Action: Run dBASE Program: DO SRCHDNR
 Before dBASE Code for this item:
 --------------------------------
   *---Open Supply database
   SELECT A
   USE PERSONNE ORDER PI
   GO TOP
   SELECT B
   USE ACCTS ORDER JON
   GO TOP
   SELECT C
   USE SUPPLY ORDER DOCNR
   SET RELATION TO PI INTO PERSONNE, JON INTO ACCTS

 After dBASE Code for this item:
 --------------------------------
   *---Close Supply database
   CLOSE DATABASES
------------------------------------------------------------

Bar: 3
 Prompt: REMOVE MARKED TRANSACTIONS
 Action: Pack Current File
 Window WINDOW4 FROM 10,10 TO 20,60 Double

 Before dBASE Code for this item:
 --------------------------------
   *---Open Supply database
   USE supply
 After dBASE Code for this item:
 --------------------------------
   *---Save changes and close database
   CLOSE DATABASES
------------------------------------------------------------

Bar: 4
 Prompt: PRINT AGING REPORT
 Action: Run Report Form AGING.frm
 Command Options:
  PLAIN
  NOEJECT
 Print Mode: Send to Default Printer
 New Database/View: AGING.QBE

 After dBASE Code for this item:
 --------------------------------
   SET CONSOLE ON
------------------------------------------------------------

Bar: 5
 Prompt: PRINT OUTSTANDING REQN REPORT
 Action: Run Report Form SUPSTAT.frm
 Command Options:
  PLAIN
  NOEJECT
 Print Mode: Send to Default Printer
 New Database/View: SUPSTAT.QBE

 After dBASE Code for this item:
 --------------------------------
   SET CONSOLE ON
------------------------------------------------------------
```

```
Bar: 6
 Prompt: PRINT SUPPLY OBLIGATION REPORT
 Action: Run Report Form SUPCHG.frm
 Command Options:
  PLAIN
  NOEJECT
 Print Mode: Send to Default Printer
 New Database/View: SUPCHG.QBE

 After dBASE Code for this item:
 --------------------------------
   SET CONSOLE ON
------------------------------------------------------------
Bar: 7
 Prompt: PRINT REQN STATUS REPORT
 Action: Run Report Form SUPRQNST.frm
 Command Options:
  PLAIN
  NOEJECT
 Print Mode: Send to Default Printer
 New Database/View: SUPRQNST.QBE

 After dBASE Code for this item:
 --------------------------------
   SET CONSOLE ON
------------------------------------------------------------

Layout Report for Popup Menu: LABMENU
-------------------------------------

Screen Image:
30        40        50        60        70
  >....+....¦....+....¦....+....¦....+....¦....+....¦....+...
05:
06:      ╔═════════════════════════════════╗
07:      ║ADD PAYROLL RECORDS              ║
08:      ║VIEW/EDIT/DELETE PAYROLL RECORDS ║
09:      ║PRINT LABOR EXPENSE REPORT       ║
10:      ║PRINT PAYRECORD REPORT           ║
11:      ╚═════════════════════════════════╝
12:
Setup for LABMENU follows:
--------------------------
 Use database/view and index file(s) in effect at run time.

Colors for Menu/Picklist:
-------------------------
 Color Settings:
   Text             : W+/N
   Heading          : W+/N
   Highlight        : GR+/B
   Box              : W+/R
   Messages        : W+/N
   Information     : B/W
   Fields          : N/BG

Bar actions for Menu LABMENU follow:
------------------------------------
Bar: 1
 Prompt: ADD PAYROLL RECORDS
 Action: APPEND
 Format File: labor.fmt
 Before dBASE Code for this item:
 ---------------------------------

   *---Open Temporary Labor database
   SELECT A
   USE PERSONNE ORDER IDCODE
   SELECT B
   USE ACCTS ORDER JON
   SELECT C
   USE TEMPLAB
   SET RELATION TO IDCODE INTO PERSONNE, JON INTO ACCTS

 After dBASE Code for this item:
 ---------------------------------

   *---Update Labor database and save
   CLOSE DATABASES
   *---Calls Payroll procedure from Acctproc file
   DO payroll
------------------------------------------------------------

Bar: 2
 Prompt: VIEW/EDIT/DELETE PAYROLL RECORDS
 Action: Run dBASE Program: DO SRCHLAB
 Use database/view and index file(s) in effect at run time.

 Before dBASE Code for this item:
 ---------------------------------
   *---Open databases
   DO payedit

 After dBASE Code for this item:
 ---------------------------------
   *---Close Labor database
   CLOSE DATABASES
   *---Call Payroll procedure from Acctproc file
   DO payroll
------------------------------------------------------------
```

```
Bar: 3
 Prompt: PRINT LABOR EXPENSE REPORT
 Action: Run Report Form LABCHGS.frm
 Command Options:
  PLAIN
  NOEJECT
 Print Mode: Send to Default Printer
 New Database/View: LABCHGS.QBE

 After dBASE Code for this item:
 --------------------------------
   set console on
------------------------------------------------------------

Bar: 4
 Prompt: PRINT PAYRECORD REPORT
 Action: Run Report Form INDPAY.frm
 Command Options:
  PLAIN
  NOEJECT
 Print Mode: Send to Default Printer
 New Database/View: PAYREC.QBE

 After dBASE Code for this item:
 --------------------------------
   SET CONSOLE ON
------------------------------------------------------------


Layout Report for Popup Menu: TRAVMENU
--------------------------------------
Screen Image:
40        50        60        70        80
>....+....¦....+....¦....+....¦....+....¦
04:
05:  ╔═══════════════════════════════╗
06:  ║ADD NEW ORDER                  ║
07:  ║VIEW/EDIT ORDERS               ║
08:  ║REMOVE MARKED ORDERS           ║
09:  ║PRINT TRAVEL OBLIGATION REPORT ║
10:  ║PRINT TRAVEL TICKLER REPORT    ║
11:  ║PRINT DELINQUENT TRAVEL CLAIM  ║
12:  ║   REPORT                      ║
13:  ║PRINT FLAG APPROVAL STATUS     ║
14:  ║   REPORT                      ║
15:  ╚═══════════════════════════════╝
16:
Setup for TRAVMENU follows:
---------------------------
 Use database/view and index file(s) in effect at run time.
```

```
Colors for Menu/Picklist:
-------------------------
 Color Settings:
   Text              : W+/N
   Heading           : W+/N
   Highlight         : GR+/B
   Box               : W+/R
   Messages          : W+/N
   Information       : B/W
   Fields            : N/BG

Bar actions for Menu TRAVMENU follow:
-------------------------------------
Bar: 1
 Prompt: ADD NEW ORDER
 Action: APPEND
 Format File: travel.fmt

 Before dBASE Code for this item:
 --------------------------------
   *---Open Travel database
   SELECT A
   USE PERSONNE CRDER PI
   GO TOP
   SELECT B
   USE ACCTS ORDER JON
   GO TOP
   SELECT C
   USE TRAVEL
   SET RELATION TO PI INTO PERSONNE, JON INTO ACCTS

 After dBASE Code for this item:
 --------------------------------
   *---Close Travel database
   CLOSE DATABASES
------------------------------------------------------------

Bar: 2
 Prompt: VIEW/EDIT ORDERS
 Action: Run dBASE Program: DO SRCHTRAV

 Before dBASE Code for this item:
 --------------------------------
   *---Open Travel databases
   SELECT A
   USE PERSONNE ORDER PI
   GO TOP
   SELECT B
   USE ACCTS ORDER JON
   GO TOP
   SELECT C
   USE TRAVEL ORDER LASTNAME
   SET RELATION TO PI INTO PERSONNE, JON INTO ACCTS

 After dBASE Code for this item:
   *---Close Travel database
   CLOSE DATABASES
------------------------------------------------------------

Bar: 3
 Prompt: REMOVE MARKED ORDERS
 Action: Pack Current File
 Window WINDOW6 FROM 10,10 TO 20,60 Double

 Before dBASE Code for this item:
 --------------------------------
   *---Open Travel database
   USE TRAVEL

 After dBASE Code for this item:
 --------------------------------
   *---Close and save changes
   CLOSE DATABASES
------------------------------------------------------------

Bar: 4
 Prompt: PRINT TRAVEL OBLIGATION REPORT
 Action: Run Report Form TRAVSTAT.frm
 Command Options:
  PLAIN
  NOEJECT
 Print Mode: Send to Default Printer
 New Database/View: TRAVSTAT.QBE

 After dBASE Code for this item:
 --------------------------------
   SET CONSOLE ON
------------------------------------------------------------

Bar: 5
 Prompt: PRINT TRAVEL TICKLER REPORT
 Action: Run Report Form TRAVPKUP.frm
 Command Options:
  PLAIN
  NOEJECT
 Print Mode: Send to Default Printer
 New Database/View: TRAVPKUP.QBE

 After dBASE Code for this item:
 --------------------------------
   SET CONSOLE ON
```

```
Bar: 6
 Prompt: PRINT DELINQUENT TRAVEL CLAIM
 Action: Run Report Form DELQTRAV.frm
 Command Options:
  PLAIN
  NOEJECT
 Print Mode: Send to Default Printer
 New Database/View: DELQTRAV.QBE

 After dBASE Code for this item:
 --------------------------------
   SET CONSOLE OFF
------------------------------------------------------------

Bar: 7
 Prompt: REPORT
 Action: Text only defined for this option - NO ACTION
------------------------------------------------------------

Bar: 8
 Prompt: PRINT FLAG APPROVAL STATUS
 Action: Run Report Form FLAGAPP.frm
 Command Options:
  PLAIN
  NOEJECT
 Print Mode: Send to Default Printer
 New Database/View: FLAGAPP.QBE

 After dBASE Code for this item:
 --------------------------------
   SET CONSOLE ON
------------------------------------------------------------

Bar: 9
 Prompt: REPORT
 Action: Text only defined for this option - NO ACTION
------------------------------------------------------------
Layout Report for Popup Menu: PROPMENU
------------------------------------
Screen Image:
50        60        70        80
>....+....|....+....|....+....|
03:
04:      ╔════════════════════════╗
05:      ║ ADD NEW PROPERTY ITEM  ║
06:      ║ VIEW/EDIT TAGGED ITEM  ║
07:      ║ REMOVE MARKED ITEMS    ║
08:      ║------------------------║
09:      ║ CHOOSE REPORT TO PRINT ║
10:      ╚════════════════════════╝
11:
```

```
Setup for PROPMENU follows:
---------------------------
 Description: Options for property module
 Message Line Prompt for Menu: Select desired property module
 option.

Colors for Menu/Picklist:
-------------------------
 Color Settings:
   Text            : W+/B
   Heading         : W+/B
   Highlight       : GR+/BG
   Box             : GR+/R
   Messages        : W+/B
   Information     : B/W
   Fields          : N/BG

Bar actions for Menu PROPMENU follow:
-------------------------------------
Bar: 1
 Prompt: ADD NEW PROPERTY ITEM
 Action: APPEND
 Format File: property.fmt

 Use database/view and index file(s) in effect at run time.

 Before dBASE Code for this item:
 --------------------------------
   *---This will call the PROPOPEN procedure to open the
   *   PERSONNE and PROPERTY databases, and relate them
   *   through the IDCODE common field.
   DO PropOpen

 After dBASE Code for this item:
 --------------------------------
   *---Closes the databases
   CLOSE DATABASES
------------------------------------------------------------

Bar: 2
 Prompt: VIEW/EDIT TAGGED ITEM
 Action: Run dBASE Program: DO SRCHTGNR
 Use database/view and index file(s) in effect at run time.

 Before dBASE Code for this item:
 --------------------------------
   *---This will  run  the  PROPOPEN  procedure  to  open  the
PERSONNE and PROPERTY
   *---databases, which are linked through the IDCODE common
fields.
   DO PropOpen
 After dBASE Code for this item:
 --------------------------------
   *---Closes the databases
   CLOSE DATABASES
------------------------------------------------------------

Bar: 3
 Prompt: REMOVE MARKED ITEMS
 Action: Pack Current File
 New Database/View: PROPERTY.DBF
 New Index File(s): PROPERTY.MDX
 New Index Order: TAGNR
------------------------------------------------------------

Bar: 4
 Prompt: ------------------------
 Action: Text only defined for this option - NO ACTION
------------------------------------------------------------

Bar: 5
 Prompt: CHOOSE REPORT TO PRINT
 Action: Open a Popup Menu Named: PROPREPT
------------------------------------------------------------

Layout Report for Popup Menu: TOOLMENU
--------------------------------------

Screen Image:
40        50        60        70        80
>....+....¦....+....¦....+....¦....+....¦
04:
05:            ╔═══════════════════════════╗
06:            ║BACKUP DATA TO DRIVE A     ║
07:            ║IMPORT SUPPLY DATA         ║
08:            ║REBUILD CORRUPTED INDEXES  ║
09:            ║RESTORE DATABASES FROM     ║
10:            ║  DRIVE A                  ║
11:            ║EXPORT ACCT/PERS DATA      ║
12:            ╚═══════════════════════════╝
13:

Setup for TOOLMENU follows:
---------------------------
 Use database/view and index file(s) in effect at run time.


Colors for Menu/Picklist:
-------------------------
 Color Settings:
   Text            : W+/N
   Heading         : W+/N
   Highlight       : GR+/B
   Box             : W+/R
   Messages        : W+/N
   Information     : B/W
   Fields          : N/BG

Bar actions for Menu TOOLMENU follow:
-------------------------------------
Bar: 1
 Prompt: BACKUP DATA TO DRIVE A
 Action: Run Dos Program - COPY *.DBF A:
------------------------------------------------------------

Bar: 2
 Prompt: IMPORT SUPPLY DATA
 Action: Run dBASE Program: DO SUPIMP
 Window IMP FROM 10,5 TO 20,75 Single
------------------------------------------------------------

Bar: 3
 Prompt: REBUILD CORRUPTED INDEXES
 Action: Run dBASE Program: DO RECRENDX
 Window RENDX FROM 10,5 TO 20,75 Double
------------------------------------------------------------

Bar: 4
 Prompt: RESTORE DATABASES FROM
 Action: Run dBASE Program: DO RESTORE
 Window REST FROM 10,5 TO 20,75 Single
------------------------------------------------------------

Bar: 5
 Prompt: DRIVE A
 Action: Text only defined for this option - NO ACTION
------------------------------------------------------------

Bar: 6
 Prompt: EXPORT ACCT/PERS DATA
 Action: Run Dos Program - COPY *.DBF B:
 Window EXP FROM 10,5 TO 20,75 Single
------------------------------------------------------------

Layout Report for Popup Menu: EXITMENU
--------------------------------------
Screen Image:
30        40        50        60        70        80
>....+....¦....+....¦....+....¦....+....¦....+....¦
04:
05:                                ╔══════════════╗
06:                                ║QUIT TO DOS   ║
07:                                ╚══════════════╝
```

```
Setup for EXITMENU follows:
---------------------------
 Use database/view and index file(s) in effect at run time.

Colors for Menu/Picklist:
-------------------------
 Color Settings:
   Text              : W+/N
   Heading           : W+/N
   Highlight         : GR+/B
   Box               : W+/R
   Messages          : W+/N
   Information       : B/W
   Fields            : N/BG

Bar actions for Menu EXITMENU follow:
-------------------------------------

Bar: 1
 Prompt: QUIT TO DOS
 Action: Quit to DOS:
------------------------------------------------------------

Layout Report for Popup Menu: DACCMENU
--------------------------------------

Screen Image:
0         10        20        30        40        50
>....+....¦....+....¦....+....¦....+....¦....+....¦
12:
13:
14:            ADD NEW ALLOCATIONS
15:            VIEW/EDIT ALLOCATIONS
16:            REMOVE MARKED ALLOCATIONS
17:            PRINT NA SUMMARY
18:
19:

Setup for DACCMENU follows:
---------------------------

Colors for Menu/Picklist:
-------------------------
 Color Settings:
   Text              : W+/N
   Heading           : W+/N
   Highlight         : GR+/B
   Box               : W+/R
   Messages          : W+/N
   Information       : B/W
   Fields            : N/BG
```

```
Bar actions for Menu DACCMENU follow:
-------------------------------------
Bar: 1
 Prompt: ADD NEW ALLOCATIONS
 Action: APPEND
 Format File: daccts.fmt

 Before dBASE Code for this item:
 --------------------------------
   *---Open Direct Account database
   SELECT A
   USE PERSONNE ORDER PI
   SELECT B
   USE ACCTS ORDER JON
   SELECT C
   USE DACCTS
   SET RELATION TO PI INTO PERSONNE, JON INTO ACCTS

 After dBASE Code for this item:
 --------------------------------
   *---Close Direct Account database
   CLOSE DATABASES
------------------------------------------------------------
Bar: 2
 Prompt: VIEW/EDIT ALLOCATIONS
 Action: Run dBASE Program: DO SRCHALOT

 Before dBASE Code for this item:
 --------------------------------
   *---Open Direct Account database
   CLOSE DATABASES
   SELECT A
   USE PERSONNE ORDER PI
   GO TOP
   SELECT B
   USE ACCTS ORDER JON
   GO TOP
   SELECT C
   USE DACCTS ORDER NAME
   SET RELATION TO PI INTO PERSONNE, JON INTO ACCTS

 After dBASE Code for this item:
 --------------------------------
   *---Close Direct Account database
   CLOSE DATABASES
------------------------------------------------------------

Bar: 3
 Prompt: REMOVE MARKED ALLOCATIONS
 Action: Pack Current File
 Window WINDOW2 FROM 10,10 TO 20,60 Double
 Before dBASE Code for this item:
 --------------------------------
   *---Open Direct Account database
   USE daccts

 After dBASE Code for this item:
 --------------------------------
   *---Save changes and close database
   CLOSE DATABASES
------------------------------------------------------------

Bar: 4
 Prompt: PRINT NA SUMMARY
 Action: Run dBASE Program: DO DIRECT

After dBASE Code for this item:
 --------------------------------
   set print off
   close databases
------------------------------------------------------------

Layout Report for Popup Menu: PROPREPT
---------------------------------------
Screen Image:

30        40        50        60        70        80
>....+....|....+....|....+....|....+....|....+....|
08:
09:
10:                    ╔═════════════════════════╗
11:                    ║*CUSTODIAN LOG           ║
12:                    ║*DISPOSAL REPORT         ║
13:                    ║*HISTORY REPORT          ║
14:                    ║*LOCATION REPORT         ║
15:                    ║*MINOR PROPERTY INVENTORY║
16:                    ║*PLANT PROPERTY INVENTORY║
17:                    ╚═════════════════════════╝
18:
Setup for PROPREPT follows:
---------------------------

 Description: SELECTION OF PROPERTY REPORTS FOR PRINTING
 Message Line Prompt for Menu: Select Property Report for
printing.

Colors for Menu/Picklist:
-------------------------
 Color Settings:
   Text            : W+/B
   Heading         : W+/B
   Highlight       : GR+/BG
   Box              : GR+/R
   Messages         : W+/B
   Information      : B/W
   Fields           : N/BG

Bar actions for Menu PROPREPT follow:
-------------------------------------
Bar: 1
 Prompt: *CUSTODIAN LOG
 Action: Run Report Form PROPCUST.frm
 Command Options:
  PLAIN
  NOEJECT
 Print Mode: Ask User at Runtime
 New Database/View: PROPCUST.QBE
------------------------------------------------------------

Bar: 2
 Prompt: *DISPOSAL REPORT
 Action: Run Report Form PROPDISP.frm
 Command Options:
  PLAIN
  NOEJECT
 Print Mode: Ask User at Runtime
 New Database/View: PROPDISP.QBE
------------------------------------------------------------

Bar: 3
 Prompt: *HISTORY REPORT
 Action: Run Report Form PROPHIST.frm
 Command Options:
  PLAIN
  NOEJECT
 Print Mode: Ask User at Runtime
 New Database/View: PROPHIST.QBE
------------------------------------------------------------

Bar: 4
 Prompt: *LOCATION REPORT
 Action: Run Report Form PROPLOCN.frm
 Command Options:
  PLAIN
  NOEJECT
 Print Mode: Ask User at Runtime
 New Database/View: PROPLOCN.QBE
------------------------------------------------------------

Bar: 5
 Prompt: *MINOR PROPERTY INVENTORY
 Action: Run Report Form PROPINVM.frm
 Command Options:
  PLAIN
  NOEJECT
 Print Mode: Ask User at Runtime
 New Database/View: PROPINVM.QBE
------------------------------------------------------------

Bar: 6
 Prompt: *PLANT PROPERTY INVENTORY
 Action: Run Report Form PROPINVP.frm
 Command Options:
  PLAIN
  NOEJECT
 Print Mode: Ask User at Runtime
 New Database/View: PROPINVP.QBE
------------------------------------------------------------

End of Application Documentation
```

APPENDIX E

FMIS 2.0 USER'S GUIDE UPDATE

A. INTRODUCTION

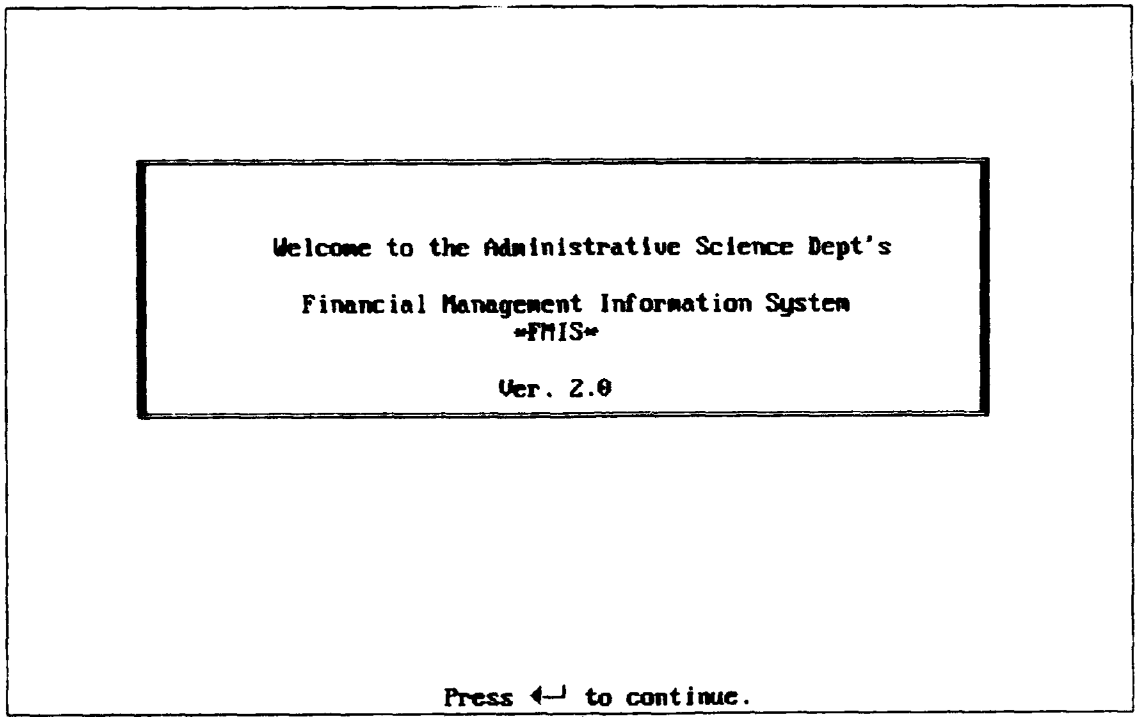
This guide is an update to the original FMIS user's manual provided as Appendix D in reference 4. This update will cover the changes to the first FMIS version and should be used in conjunction with the original user's guide. The major change reflected in FMIS version 2.0 is the addition of a Property Management System (PMS). The new system was integrated into the original program utilizing the existing system architecture resulting in little change to the program interface. Users familiar with operating the original FMIS should have no problem operating FMIS 2.0. Installation of the system should be in accordance with instruction provided in reference 4.

B. STORAGE REQUIREMENTS

The incorporation of the PMS into FMIS 2.0 results in slightly larger storage requirements than the original system. The initial FMIS program without data files was approximately 1.04 MB and could be stored on a single high density 5 1/4" floppy disk. Version 2.0 at 1.42 MB requires two 5 1/4" high density disks or a single 1.44 MB high density 3 1/2" disk. A copy of the original system disk should always be kept in a safe place in case system restoration is required. Hard disk requirements for the entire application, including dBASE program and data file storage, still require a minimum of approximately 10 MB.

C. OPERATION

Once the FMIS program is started in the normal manner, the "Welcome Screen" will appear. The screen should be identical to that shown in Figure E.1. If there is a discrepancy, the user should check to ensure version 2.0 is the current program being executed.



Welcome to the Administrative Science Dept's

Financial Management Information System
FMIS

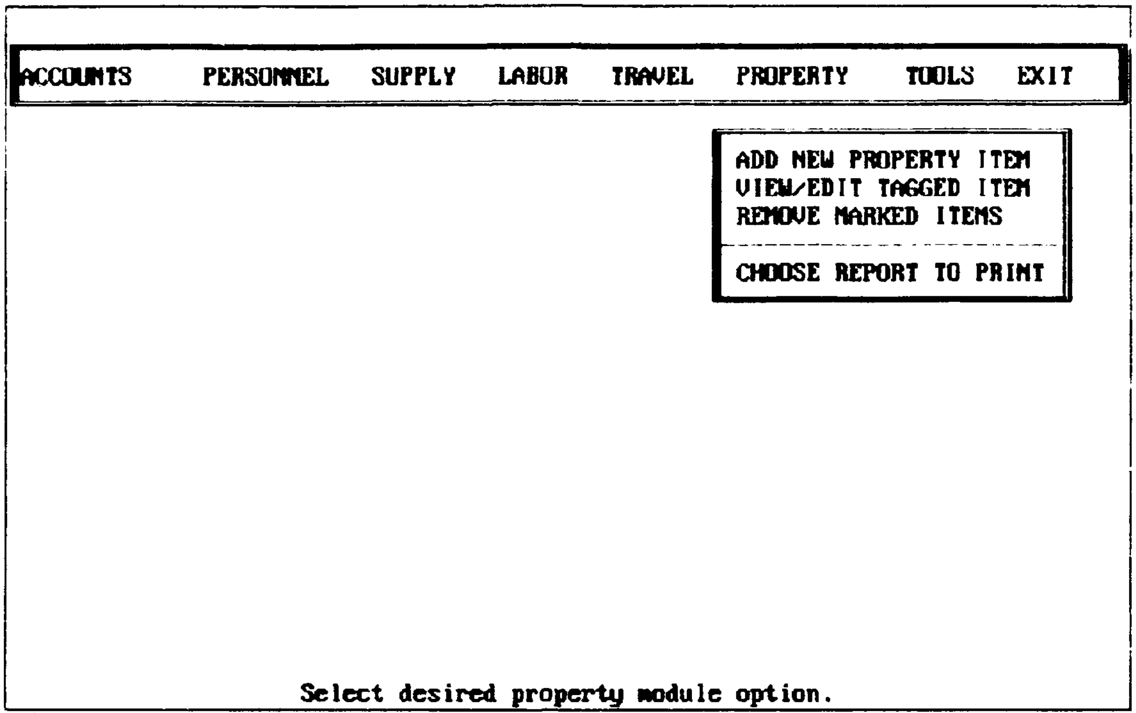
Ver. 2.0

Press ↵ to continue.

Figure E.1 FMIS 2.0 Introduction Screen

D. MAIN MENU

The only noticeable change to the main menu is the addition of the PROPERTY module located to the right of the TRAVEL option (see Figure E.2). Access the desired option



Select desired property module option.

Figure E.2 Main Menu / Property Pop-Up Menu

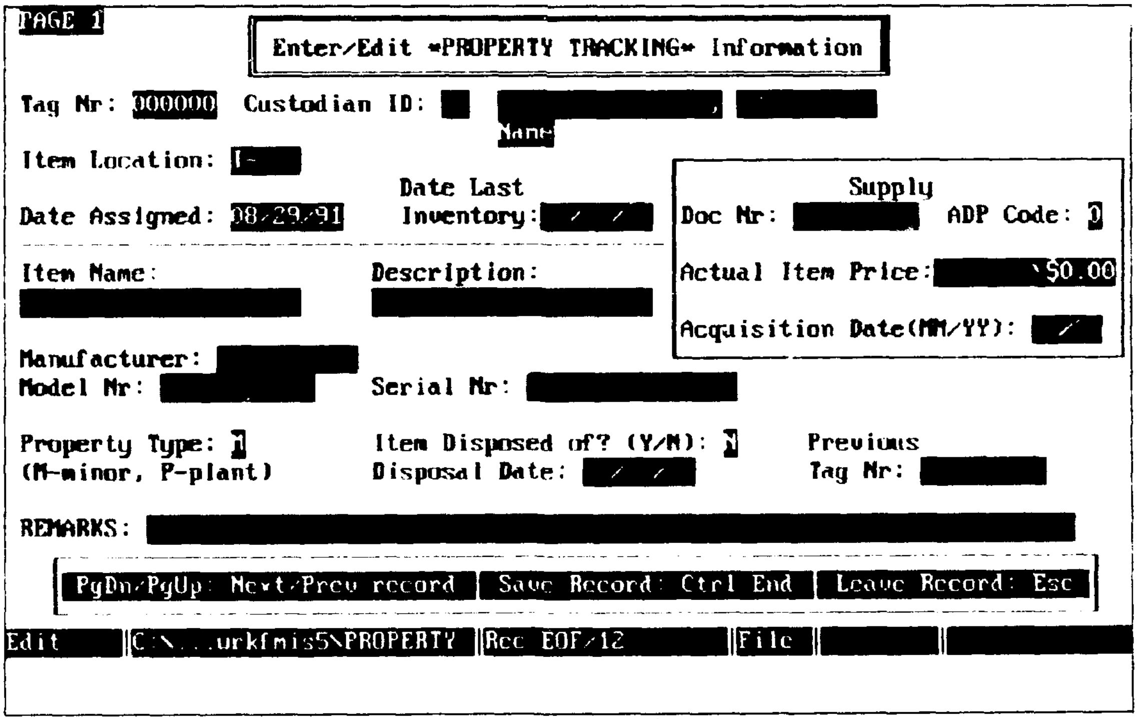
by pressing the left or right arrow keys. As each successive module is highlighted the associated pop-up menu will appear.

E. THE PROPERTY MODULE

The PROPERTY module is used for management and tracking of accountable minor and plant property assigned in the A.S. Department. It provides a method to maintain an active database of the tagged property item's assigned custodian, location, cost data, classification information, and historical data. Selections included in the PROPERTY pop-up menu allow for the entry of new records, specific record editing or viewing, deletion of marked records, and report printing. The PROPERTY module links to the PERSONNEL module for retrieval of current custodian information.

1. Add New Property Item

To add a new property record, select "Add New Property Item" from the PROPERTY menu. To execute this option use the arrow keys to move the highlight to this choice and press <Enter>. Choosing this selection will clear the current screen and a formatted data entry screen (Figure E.3) will appear. The screen should display an empty record with the cursor in the Tag Number field. Each property record input form consists of two pages. Check the upper left corner to verify that you are on page one. Several fields have unique features which will be explained below. Note, fields accepting data entry are colored blue, Gray fields only display information and do not require entry.



PAGE 1

Enter/Edit *PROPERTY TRACKING* Information

Tag Nr: 000000 Custodian ID: , Name

Item Location: I-

Date Assigned: 08/29/91 Date Last Inventory: / /

Supply
Doc Nr: ADP Code:
Actual Item Price: $0.00
Acquisition Date(MM/YY): /

Item Name: Description:

Manufacturer:
Model Nr: Serial Nr:

Property Type: (M-minor, P-plant)
Item Disposed of? (Y/N):
Disposal Date: / /
Previous Tag Nr:

REMARKS:

Figure E.3 Property Screen, Page One

a. *Entry Screen Page One*

Enter the Tag Number of the new property item to be tracked. The system will verify the entry to ensure that you are not entering a number already assigned to another piece of property, sounding a beep and displaying a message to enter a new number if this occurs. The field will automatically right justify the number so you may enter it without the leading zeros.

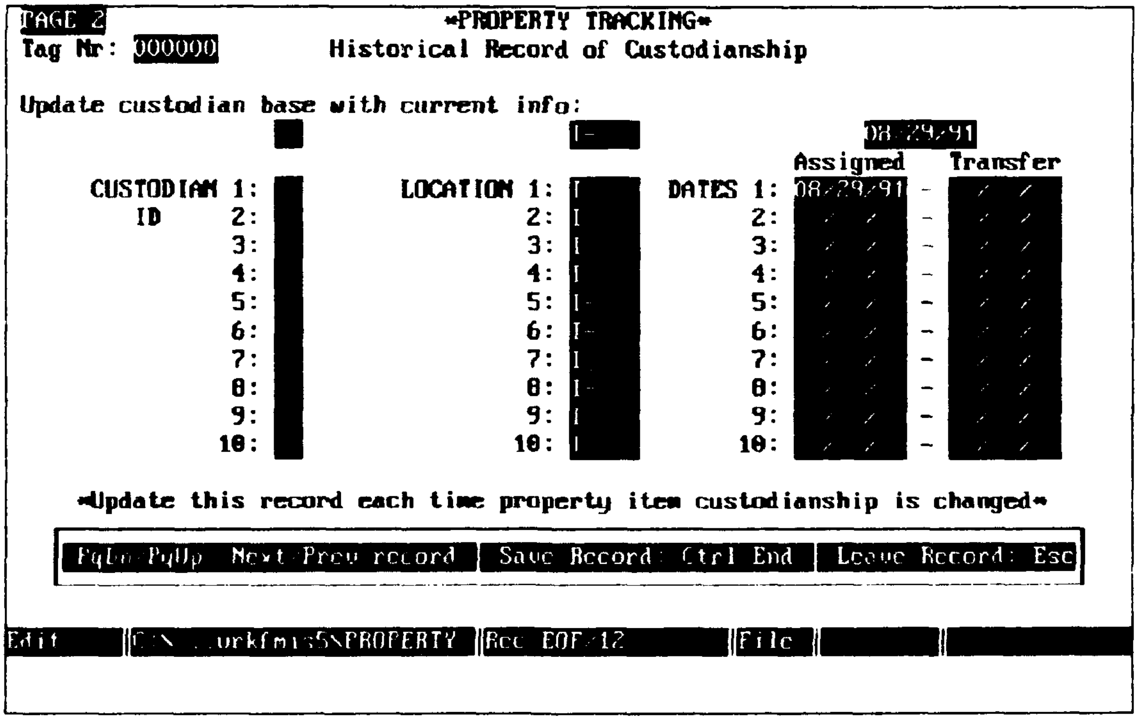
Enter the two character personnel ID in the Custodian ID field. The entry will be validated against personnel codes in the personnel file. If a valid code was entered, the corresponding name will automatically appear in the adjacent display field. If an invalid code was entered the system will beep and an error message requesting you try again will appear.

The "I-" appears in the location box as a default since most A.S. department property will be located in Ingersoll Hall. Simply fill in the room number, or if desired the building character can be overwritten. A default value of "0" appears in the ADP Code field for non-ADP property, it can be written over if an ADP code (1-5) is desired. Numbers greater than "5" are invalid and will not be accepted. The Property Type field will have a default value of "M" for minor property. The Item Disposed Of field will default to "N" (no). Until this is changed to "Y", the Disposal Date field will be bypassed.

Upon pressing the <Tab> or <Enter> keys at the Remarks field the screen will advance to page two of the entry form. The <PgDn> key can alsc be used to advance to page two.

b. Entry Screen Page Two

The second page of the entry screen (Figure E.4) is used to enter historical tracking data for the tagged property item. When adding a new property item no operator entry is necessary. The system will automatically fill the first custodian, first location, and first date fields by copying the related data entered by the operator on the first page. Current data from the first screen is shown in flashing fields to verify the entries in the historical fields.



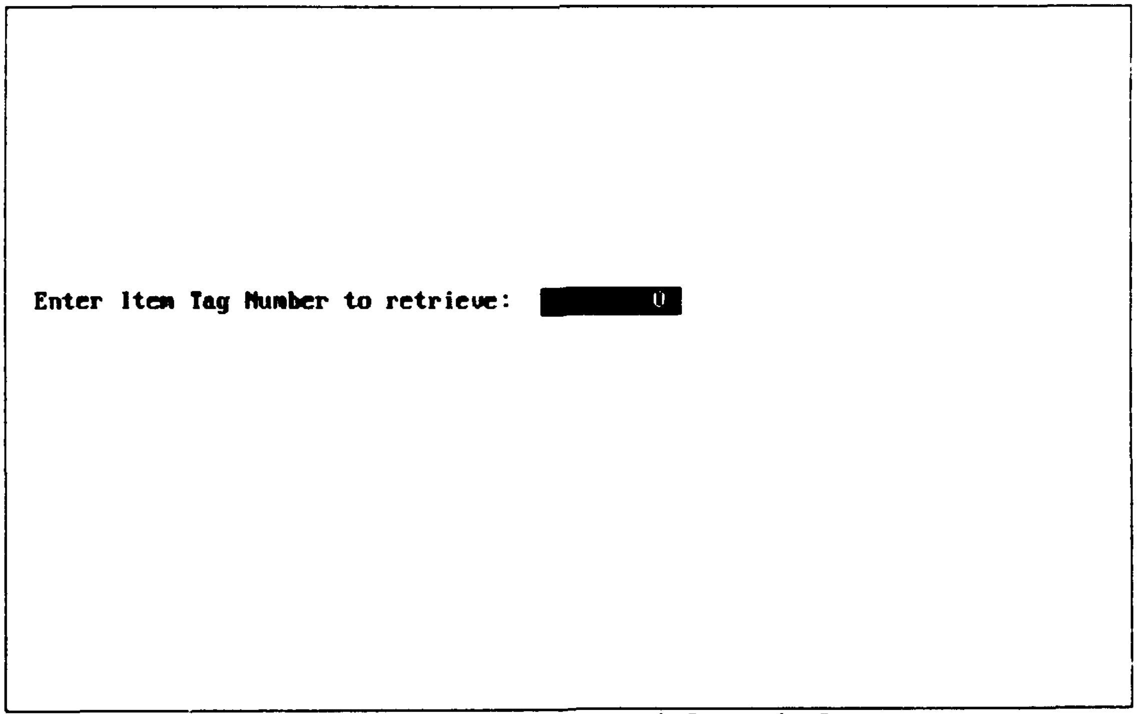
```
PAGE 2                        *PROPERTY TRACKING*
Tag Nr: 000000          Historical Record of Custodianship

Update custodian base with current info:
                                  I-                 08/29/91
                                                  Assigned    Transfer
   CUSTODIAN 1:           LOCATION 1: I       DATES 1: 08/29/91 -   /  /
      ID     2:                    2: I             2:   /  /   -   /  /
             3:                    3: I             3:   /  /   -   /  /
             4:                    4: I             4:   /  /   -   /  /
             5:                    5: I-            5:   /  /   -   /  /
             6:                    6: I-            6:   /  /   -   /  /
             7:                    7: I             7:   /  /   -   /  /
             8:                    8: I-            8:   /  /   -   /  /
             9:                    9: I             9:   /  /   -   /  /
            10:                   10: I            10:   /  /   -   /  /

  *Update this record each time property item custodianship is changed*

 PgDn/PgUp  Next/Prev record | Save Record: Ctrl End | Leave Record: Esc

Edit     ||C:\ ..urkfmis5\PROPERTY ||Rec EOF/12        ||File ||       ||
```

Figure E.4 Property Screen Page Two

To enter another new record press <PgDn> to obtain a clear screen. To save newly entered records and return to the Main Menu press the <Ctrl><End> keys simultaneously. Pressing the <Esc> key will abandon the new record and return to the Main Menu without saving it.

2. View/Edit Tagged Item

To view or edit a specific tagged property item choose this option from the Property pop-up menu. Once this selection is made, the search screen shown in Figure E.5 will be displayed. This screen queries the user to input the Tag Number of the desired property record to be viewed or edited. The entered number will be justified to the right, so the user doesn't need to input the leading zeros. If a valid tag number (number corresponding to an existing record) is entered



Enter Item Tag Number to retrieve: 0

Figure 5Figure E.6 Property Record Search Screen

the system will go directly to that record and display it's data. If the tag number is not valid, a message will appear asking the user to try again. Pressing the <Enter> key with no input will cause the system to display the property record with the lowest value tag number.

The Enter/Edit screens shown previously in Figures E.3 and E.4 will display the requested record. The arrangement of the fields is designed to permit routine editing with a minimum of keystrokes. Normal change of custodian editing will consist of entering modifications in the Custodian ID, Location, and Date Assigned fields all located above the dotted line on the edit screen. When changing custodian information be sure to advance to page two and update the historical data by filling in the next available line with the information supplied in the flashing display fields. Editing can be aborted with no changes to the original record data by pressing <Esc> before advancing to the next record (advancing to the next record using PgDn will save changes). To save changes and return to the main menu press <Ctrl><End>. New records may not be added from any edit screen in the system.

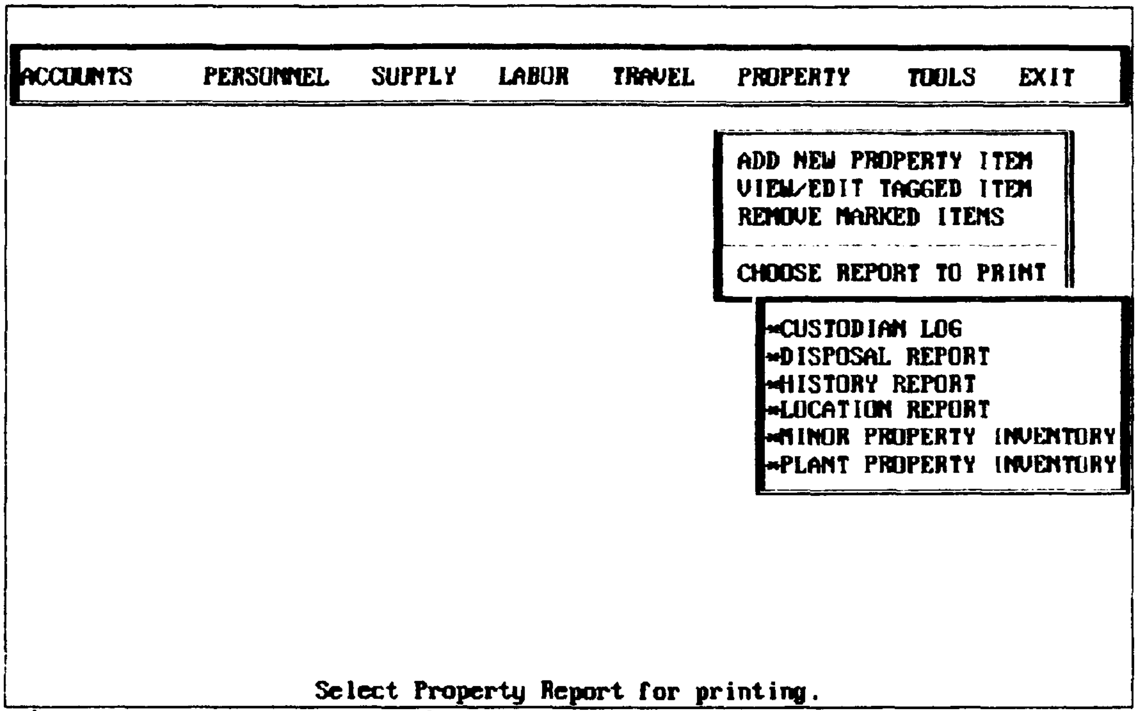
Deleting a property record follows the same procedures used to delete records in other FMIS modules. If you are displaying the record you wish to delete, press <Ctrl><U> to mark the record. Marking does not interfere with further viewing or editing operations. Pressing <Ctrl><End> saves the deletion mark as well as any editing changes. Returning to the marked record and pressing <Ctrl><U> again will unmark the record. A marked record is not actually deleted until the "Remove Marked Items" option is chosen from the Property pop-up menu.

3. **Remove Marked Items**

This selection in the pop-up menu performs the final record deletion. After completing this option, the marked record(s) will be permanently removed from the database. Extreme care should be taken when deleting records. Since the system was designed to maintain property records even after disposal, record deletions will not be performed routinely. After a record deletion the screen will scroll various file specifications as the system automatically re-sorts and indexes the database. Once this is complete the system will return to the main menu.

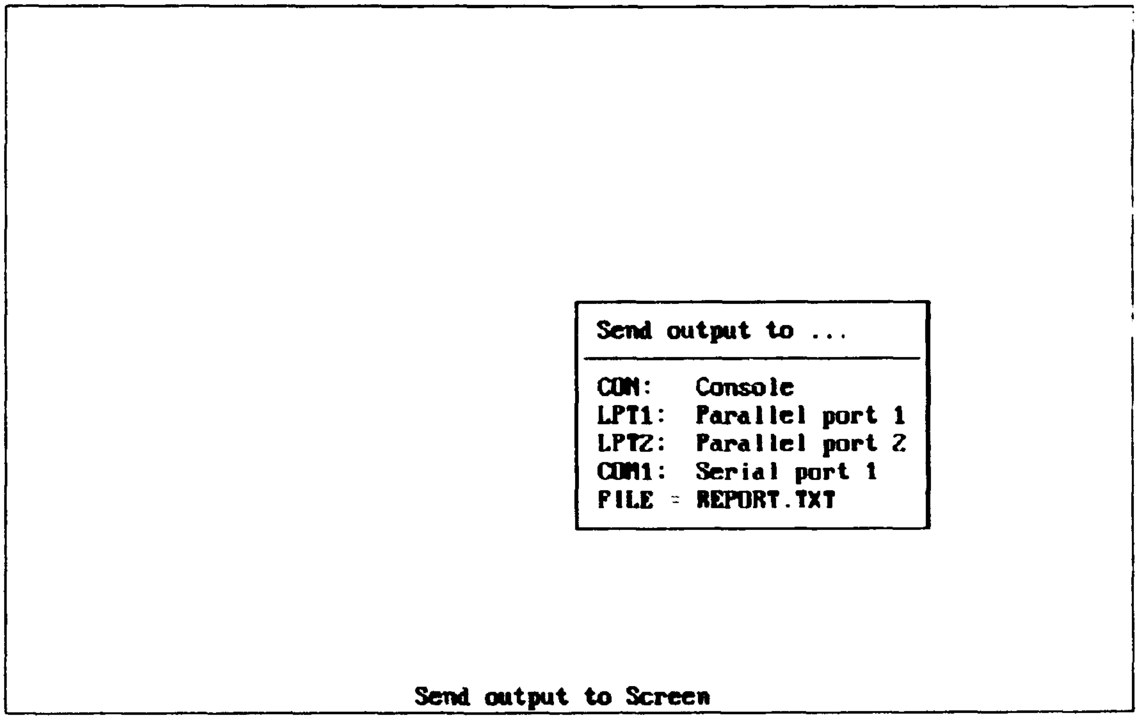
4. **Choose Report to Print**

When highlighting this option and pressing <Enter> a pull-down menu (Figure E.6) will appear offering a choice of the six various reports described below. To choose one, simply highlight the desired report and <Enter>. This will invoke a final menu (Figure E.7) allowing the user to direct the output to the screen console, a printer (LPT1 or 2), or to a file. The normal choice will usually be the LPT1 printer port to produce a hard copy. To move back up the menu hierarchy to the Main Menu press <Esc>.



Select Property Report for printing.

Fi3ure E.6 Report Pull-down Menu



Send output to Screen

Figure E.7 ?eport Destination Menu

a. Custody Log

This general purpose report lists all current tagged property in the A.S. department, sorted in alphabetical order by the property custodian's last name and ID code. A cost subtotal for all property assigned to each custodian is provided. A grand total of all tagged property tracked in the system is provided as well, with the exception of disposed property which is not reflected in this report.

b. Disposal Report

The Disposal Report lists all recorded disposed of property. These items are grouped by ADP code (0-5), with a subtotal for each disposed ADP group. A grand total is computed for all disposed of property.

c. History Report

This report lists historical custodian, location, and possession dates for each property item in tag number sequence.

d. Locatic Report

This report provides the same basic information as the custody log. However, current property items are grouped by location with custodians subgrouped within each location. Subtotals are provided for each location and a grand total for all current tagged items.

e. Minor Property Inventory

All current minor property is listed in tag number sequence along with the current custodian, location, and latest inventory date. A handy block for writing dates and initials makes this report useful when conducting visual inventories.

f. Plant Property Inventory

This report is identical to the previous one except that it lists only current plant property items.

F. TOOLS AND OTHER FMIS FUNCTIONS

Operation of the tool utilities and other module functions provided by FMIS 2.0 are identical to the original FMIS. If a review of any of these procedures is necessary, refer to the original FMIS user's guide provided as Appendix D in Reference 4.

LIST OF REFERENCES

1. Renner, R.B., *Information Requirements Analysis: An Application*, Master's Thesis, Naval Postgraduate School, Monterey, CA, March 1984.

2. Booker, R.L., *The AS Financial Reporting System: Some Experience On Prototyping And User Interaction*, Master's Thesis, Naval Postgraduate School, Monterey, CA, March 1986.

3. Sexton, T.M., *A Property Management System For The Administrative Sciences Department*, Master's Thesis, Naval Postgraduate School, Monterey, CA, September 1987.

4. Ford, N.S., and Zimmon, N.W., *A Data-Based Financial Management Information System (FMIS) For Administrative Sciences Department*, Master's Thesis, Naval Postgraduate School, Monterey, CA, December 1990.

5. Rochkind, M., "The Source Code Control System," *IEEE Trans. Software Engineering*, vol. 1, nol. 4, December 1975.

6. Dolan, K.A., and Kroenke, D.M., *Database Processing; Fundamentals, Design, and Implementation*, Third edition, Macmillan Publishing Co.,1988.

BIBLIOGRAPHY

1. Page-Jones, M., *Practical Guide to Structured Systems Design*, 2nd ed., Prentice-Hall, Inc., 1988.

2. Senn, J.A., *Analysis and Design of Information Systems*, 2nd ed., McGraw-Hill, Inc., 1989.

3. Simpson, A., *dBASE IV Programmer's Instant Reference*, Sybex, Inc., 1989.

4. Simpson, A., *Understanding dBASE IV 1.1*, 2nd ed., Sybex, Inc., 1990.

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center 2
   Cameron Station
   Alexandria, VA 22304-6145

2. Library, Code 52 2
   Naval Postgraduate School
   Monterey, CA 93943-5002

3. Computer Technology Programs, Code 37 2
   Naval Postgraduate School
   Monterey, CA 93943-5000

4. Department of Administrative Sciences, Code AS 2
   Naval Postgraduate School
   Monterey, CA 93943-5002

5. Professor Tung Bui, Code ASBD 2
   Naval Postgraduate School
   Monterey, CA 93943-5002

6. Professor Shu Liao, Code ASLC 2
   Naval Postgraduate School
   Monterey, CA 93943-5002

7. LT. Thomas A. Ditri 1
   11292 Kensington Rd.
   Rossmoor, CA 90720